

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky  
a komunikačních technologií

BAKALÁŘSKÁ PRÁCE



# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

## FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

## ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

DEPARTMENT OF CONTROL AND INSTRUMENTATION

## 3D MODEL

3D MODEL

## BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

## AUTOR PRÁCE

AUTHOR

Martin Sládeček

## VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Miloslav Richter, Ph.D.

BRNO 2017

# Bakalářská práce

bakalářský studijní obor **Automatizační a měřicí technika**

Ústav automatizace a měřicí techniky

**Student:** Martin Sládeček

**ID:** 173740

**Ročník:** 3

**Akademický rok:** 2016/17

**NÁZEV TÉMATU:**

## 3D model

### POKYNY PRO VYPRACOVÁNÍ:

Vytvořte 3D model za pomoci pomalu se pohybující kamery. Zpracujte data získaná z kamery a vhodně je reprezentujte.

- 1) Seznamte se s danou problematikou a proved'te rozbor různých variant řešení úlohy.
- 2) Nasnímejte vzorové scény pro zpracování a testování navržených algoritmů 3D rekonstrukce objektů (mimo podstavy).
- 3) Na snímcích nalezněte významné body, nalezněte stejné body na různých snímcích pořízených z různých míst. Stanovte polohu bodů v prostoru.
- 4) Výsledek vhodně reprezentujte. Zhodnoťte dosažené výsledky.

### DOPORUČENÁ LITERATURA:

Faugeras O.: Three-Dimensional Computer Vision, The MIT Press 1993

Kraus K.: Photogrammetrie 1 und 2, Ummeler / Bonn, 1996

**Termín zadání:** 6.2.2017

**Termín odevzdání:** 29.5.2017

**Vedoucí práce:** Ing. Miloslav Richter, Ph.D.

**Konzultant:**

**doc. Ing. Václav Jirsík, CSc.**  
*předseda oborové rady*

### UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

## ABSTRAKT

Tato práce se zabývá problematikou rekonstrukce trojrozměrné scény ze snímků záznamu z obyčejné kamery. První část práce popisuje principy vyžívané při řešení úlohy, druhá část popisuje algoritmus rekonstrukce a jeho implementaci v podobě programu v jazyce Python. Tento program je dále demonstrován na několika vybraných scénách. Závěr diskutuje o kvalitě vytvořených modelů, nedostatcích programu a možných vylepšeníh.

## KLÍČOVÁ SLOVA

počítačové vidění, klíčové body, deskriptory, SIFT, SURF, 3d rekonstrukce, structure from motion, epipolární geometrie, koplanární omezení, fundamentální matice, esenciální matice, pinhole kamera, Minkowského vzdálenost, Lp norma, RANSAC, triangulace

## ABSTRACT

This paper concerns the task of three-dimensional scene reconstruction using image sequence obtained with an ordinary camera. First portion of this thesis outlines the principles used in solving of the task, the second chapter describes a reconstucion algorithm and it's implementation in the Python programming language. This implementation is then demonstrated on several selected scenes. Final remarks discuss the quality of resulting models, shortcomings of the program and possible improvements.

## KEYWORDS

computer vision, keypoints, descriptors, SIFT, SURF, 3d reconstruction, structure from motion, epipolar geometry, coplanarity constraint, fundamental matrix, essential matrix, pinhole camera, Minkowki distance, Lp norm, RANSAC, triangulation

SLÁDEČEK, Martin. *3D model*. Brno, Rok, 53 s. Bakalářská práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřicí techniky. Vedoucí práce: Ing. Miloslav Richter, Ph.D.



## PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „3D model“ jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno .....

.....

Martin Sládeček

## PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu bakalářské práce panu Ing. Miloslavu Richterovi, Ph.D. za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

Brno .....

.....

Martin Sládeček

# OBSAH

Úvod	10
<b>1 Teoretická část</b>	<b>11</b>
1.1 Detekce klíčových bodů . . . . .	11
1.1.1 Obecná charakteristika detektoru . . . . .	11
1.1.2 SIFT - Scale-invariant feature transform . . . . .	11
1.1.3 SURF - Speeded up robust features . . . . .	17
1.2 Korespondence klíčových bodů . . . . .	20
1.2.1 Detekce falešných korespondencí pomocí poměru Minkowského vzdáleností . . . . .	21
1.3 Prostorová rekonstrukce scény . . . . .	21
1.3.1 Model kamery . . . . .	21
1.3.2 Model dvou pohledů . . . . .	23
1.3.3 Postupné skládání mnoha pohledů . . . . .	29
1.3.4 Matematická formulace pro $N$ bodů a $M$ pohledů . . . . .	30
<b>2 Praktická část</b>	<b>32</b>
2.1 Volba nástrojů . . . . .	32
2.2 Algoritmus rekonstrukce . . . . .	33
2.3 Struktura programu . . . . .	33
2.3.1 Vstupní data . . . . .	34
2.3.2 Přehled důležitých operací . . . . .	35
2.3.3 Vnitřní reprezentace dat . . . . .	40
<b>3 Demonstrační scény</b>	<b>41</b>
<b>4 Závěr</b>	<b>44</b>
<b>Literatura</b>	<b>46</b>
<b>Seznam příloh</b>	<b>48</b>
A Vliv bundle adjustmentu na reprojekční chyby	50
B Reprojekce válcové nádoby do původních pohledů	51
C Rotační vektor a matice	52
D Obsah DVD	53

# SEZNAM OBRÁZKŮ

1.1	DoG pyramida . . . . .	12
1.2	Vliv volby $\sigma$ na opakovatelnost nálezu klíčového bodu mezi netransformovaným a transformovaným obrázkem podle [1] . . . . .	12
1.3	Okolí bodu $X$ , ze kterého je ověřováno maximum/minimum v DoG space . . . . .	13
1.4	Ilustrace aproximace opravdové polohy extrému v okolí bodu $X_0$ Taylorovou řadou . . . . .	14
1.5	Vážení orientací gradientu gaussovým filtrem . . . . .	15
1.6	Histogram hodnot $\theta(x, y)$ . . . . .	16
1.7	Ilustrace způsobu získání deskriptoru . . . . .	16
1.8	(vlevo)Diskrétní druhá derivace Gaussiánu ve směrech $yy$ a $xy$ a jejich aproximace (vpravo) . . . . .	18
1.9	Podoblast integrálního obrazu ohraničená body $a, b, c, d$ . . . . .	18
1.10	Kruhová výseč určující orientaci klíčového bodu SURF . . . . .	19
1.11	Pinhole camera . . . . .	21
1.12	Model dvou pohledů . . . . .	23
1.13	Epipolární geometrie . . . . .	24
1.14	(a)nejednoznačnost měřítka, (b)projektivní nejednoznačnost [8] . . . . .	25
1.15	Možné kombinace projekčních matic získaných singulárním rozkladem esenciální matice [8] . . . . .	26
1.16	Ideální a reálný případ triangulace . . . . .	29
1.17	Možné výsledky triangulace při malých chybách určení projekčních paprsků . . . . .	29
2.1	Korespondující páry klíčových bodů před a po filtraci . . . . .	36
2.2	Mrak bodů a jejich zscore . . . . .	37
2.3	Ukázka řídké matice bundle adjustmentu . . . . .	38
2.4	Vliv bundle adjustmentu na součet reprojekčních chyb v každém pohledu . . . . .	39
2.5	Důsledek bundle adjustmentu na rekonstrukci scény . . . . .	39
3.1	Scéna s knihami a její mrak bodů . . . . .	41
3.2	Scéna s krbem a její mrak bodů . . . . .	42
3.3	Scéna s válcovou nádobou a její mrak bodů . . . . .	43
4.1	Porovnání hustoty mraků získaných z různých druhů klíčových bodů . . . . .	44
A.1	Důsledek bundle adjustmentu na reprojekční chyby . . . . .	50
B.1	Reprojekce válcové nádoby do původních pohledů . . . . .	51

## SEZNAM TABULEK

2.1	Verze a stručné popisy použitých knihoven . . . . .	32
2.2	Souhrn vizualizačních metod třídy <b>Reconstruction</b> . . . . .	34

## SEZNAM VÝPISŮ

2.1	Generátor záznamů pozorování . . . . .	35
2.2	Ukázka hlavičky datového rámce <code>Reconstruction.observations</code> . .	40
2.3	Ukázka hlavičky datového rámce <code>Reconstruction.pts3d</code> . . . . .	40
2.4	Ukázka hlavičky datového rámce <code>Reconstruction.colors</code> . . . . .	40

# ÚVOD

Rekonstrukce reálné trojrozměrné struktury objektů je nedílnou součástí mnoha vědeckých a průmyslových odvětví. Využívá se jako součást v celém spektru aplikací, od lékařských diagnostických nástrojů po digitální archivaci archeologických nálezů[17][20]. Většina těchto aplikací však vyžaduje ke své funkci speciální snímací systémy jako jsou laserové skenery nebo stereoskopické kamery. Prospekt prostorové rekonstrukce pomocí standardních běžně dostupných digitálních kamer je proto velmi atraktivní.

Zadání úlohy prostorové rekonstrukce z prostých obrazových dat je pro člověka snadno představitelné, jelikož podobný proces provádíme bezmyšlenkovitě v každodenním životě při vyhodnocování zrakových vjemů. Řešení pomocí výpočetní techniky však vyžaduje důkladnou analýzu dostupných dat a vztahů, které z nich lze vyvodit. Je třeba znát:

- Jaké informace o scéně jsme schopni získat.
- Vhodné způsoby reprezentace těchto informací z hlediska dalšího zpracování.
- Vhodné způsoby modelování obrazových dat v kontextu trojrozměrné scény.
- Techniky detekce chybných a nedůležitých dat.

Všechny navržené postupy musí brát v potaz nejistoty naměřených dat a nejistoty metod použitých pro výpočty.

Odměnou za robustní návrh metody mohou být trojrozměrné modely geografického terénu[18], měst, budov[19], ale i menších objektů a interiérových prostor, třeba pro autonomní robotickou navigaci[20].

Tato práce se zaměřuje na rekonstrukci z videozáznamu digitální kamery a pracuje s následujícími předpoklady:

- Sousedním snímkům v záznamu odpovídají sousední polohy kamer v prostoru.
- Způsob snímání a parametry snímače jsou pro všechny snímky stejné.
- Záznam byl pořízen záměrně tak, aby z něj bylo možné sestavit rekonstrukci.

Cílem práce je vytvořit program pro demonstraci principů rekonstrukce z mnoha pohledů. První část této práce se věnuje teoretickým podkladům a principům nezbytným pro řešení úlohy, druhá část popisuje praktickou implementaci rekonstrukčního programu. V závěru je zhodnocení výsledků a diskuze o možných rozšířeních a vylepšeních.

# 1 TEORETICKÁ ČÁST

## 1.1 Detekce klíčových bodů

### 1.1.1 Obecná charakteristika detektoru

Algoritmus pro hledání klíčových bodů (*keypoints*) musí pro korektní a praktické fungování produkovat klíčové body, které budou významné z hlediska tvaru objektů ve scéně, půjdou efektivně vyhledat v různých snímcích a tyto body pak popsat pokud možno unikátním způsobem. Tento způsob musí být dostatečně obecný na to, aby popis (*deskriptor*) klíčového bodu byl rozpoznatelný i ve snímku s částečně transformovanou scénou (posun, natočení), a zároveň tak konkrétní, aby nedocházelo k nesprávné identifikaci s jiným klíčovým bodem.

Objekty ve scéně jsou na různých snímcích snímány pod různým natočením, jejich popis musí toto být schopný kompenzovat, tedy být rotačně invariantní (*rotation invariance*), dále musíme uvažovat, že při různých vzdálenostech kamery od objektu budou význačné (pro nás zajímavé) detaily objektu zabírat různou plochu snímku (různý počet pixelů). Deskriptor by měl tedy být pokud možno nezávislý na zvětšení (*scale invariance*). U některých aplikací musíme vzít v potaz možnost, že jednotlivé snímky mohou být získány v podmínkách s různým osvětlením a dalšími vlivy okolí. Popis bodu by neměl pracovat s absolutními hodnotami jasových úrovní snímku, ale raději s jejich normalizovanou hodnotou nebo s agregací rozdílů hodnot v dané části snímku.

### 1.1.2 SIFT - Scale-invariant feature transform

*SIFT* je metoda, kterou navrhl David Lowe v roce 1999. Patří mezi nejznámější metody pro detekci a popis klíčových bodů.

#### Výběr extrémů ve scale space

Algoritmus SIFT využívá pro vyhledání potenciálně zajímavých bodů takzvané scale space. Scale space je objekt o dimenzi větší než původní snímek. Tuto dimenzi nazýváme *scale* (značeno  $\sigma$ ). Scale space je tvořen navrstvením transformací původního snímku, kde každá následující vrstva je transformací předchozí vrstvy Gaussovým filtrem. Scale space  $L(x, y, \sigma)$  snímku s obrazovou funkcí  $I(x, y)$  vypočítáme jako

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \quad (1.1)$$

kde  $*$  je operátor konvoluce.

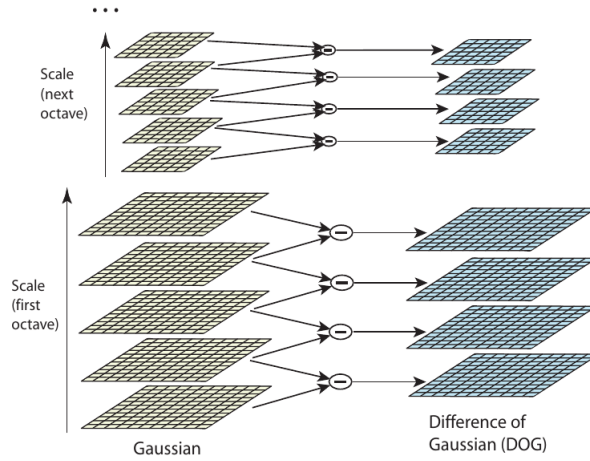


SIFT konkrétně využívá analýzu rozdílů gaussianů (Difference of Gaussians) lišících se o konstantní faktor  $k$ .

$$D(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) = L(x, y, k\sigma) - L(x, y, \sigma) \quad (1.2)$$

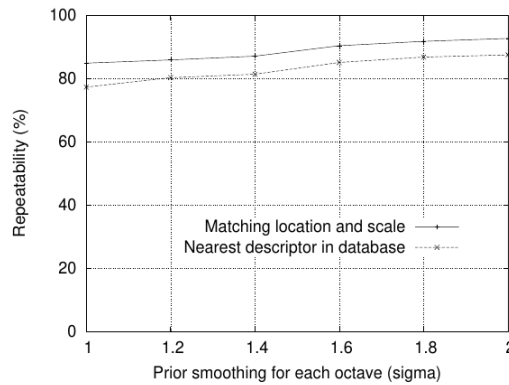
$D(x, y, \sigma)$  nazýváme Difference of Gaussians space, neboli DoG space. Důvodem pro využití DoG space je, že význačný bod sejmutý pod různými úhly se promítne do stejné pozice v DoG space.

Po vytvoření DoG space zmenšíme rozměry původního snímku na polovinu (zahodíme každý druhý řádek a každý druhý sloupec) a celý postup zopakujeme.



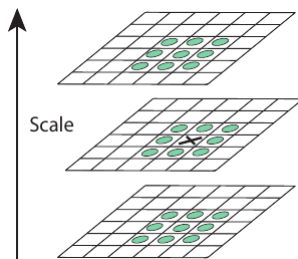
Obr. 1.1: DoG pyramida

Autor algoritmu doporučuje empiricky zjištěnou hodnotu  $\sigma \geq 1.6$  prvotního Gaussova filtrování zvětšující se s hodnotou  $k = \sqrt{2}$  pro každou následující vrstvu.[1]



Obr. 1.2: Vliv volby  $\sigma$  na opakovatelnost nálezů klíčového bodu mezi netransformovaným a transformovaným obrázkem podle [1]

Můžeme si všimnout, že s takto zvolenými hodnotami efekt Gaussova filtrování na  $k\sigma$  bude stejný jako na obrázku o polovičním rozlišení. V dalším kroku postupně projdeme každý bod v DoG space a porovnáme jeho hodnotu s hodnotami jeho nejbližších sousedů jak v jeho vrstvě tak ve vrstvách nad i pod ním.



Obr. 1.3: Okolí bodu X, ze kterého je ověřováno maximum/minimum v DoG space

Pokud je prověřovaný bod maximem nebo minimem z těchto dvaceti sedmi hodnot, označíme jej za kandidáta na klíčový bod. Může se zdát že porovnávat bod se všemi jeho sousedy bude velmi výpočetně náročné, ale v realitě můžeme potenciálního kandidáta vyřadit již po několika málo operacích.

### Filtrování kandidátů

Nyní máme mnoho kandidátů na možné klíčové body, ale tito kandidáti nejsou nezbytně vhodní, jelikož mohou mít například malý kontrast vzhledem ke svému okolí nebo mohou ležet na hraně snímaného objektu. Z hlediska výběru klíčových bodů nás nejvíce zajímají body ležící na rozích.

Je třeba si uvědomit, že nám známá diskrétní obrazová funkce uvádí hodnoty jasu pouze v jistých vzorkovaných bodech. Opravdová maxima se ale mohou nacházet nejen ve vzorkovaných bodech, ale s větší pravděpodobností v jejich bezprostředním mezipixelovém okolí. Mohlo by tedy dojít k tomu, že některá maxima by byla nedetekována na zvoleném rozlišení. Pro odfiltrování hodnot příliš nízkého kontrastu budeme muset využít hodnoty obrazové funkce mezi nám známými pixely. Tyto hodnoty určíme Taylorovou aproximací druhého řádu v nejbližším  $3 \times 3$  okolí kandidáta na klíčový bod.

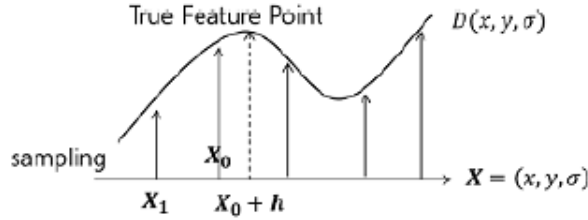
Taylorova aproximace bude mít tvar

$$\mathbf{D}(\mathbf{x}) = \mathbf{D} + \frac{\partial \mathbf{D}^T}{\partial \mathbf{x}} \mathbf{x} + \frac{1}{2} \mathbf{x}^T \frac{\partial^2 \mathbf{D}}{\partial \mathbf{x}^2} \mathbf{x} \quad (1.3)$$

kde  $\mathbf{x} = (x, y, \sigma)^T$  určuje souřadnice bodu v DoG space.

Derivace obrazové se spočítají jako difference hodnot vedlejších pixelů. Extrém této aproximace má souřadnice

$$\hat{\mathbf{x}} = -\frac{\partial^2 \mathbf{D}}{\partial \mathbf{x}^2}^{-1} \frac{\partial \mathbf{D}}{\partial \mathbf{x}}$$



Obr. 1.4: Ilustrace aproximace opravdové polohy extrému v okolí bodu  $X_0$  Taylorovou řadou

Pokud hodnota  $\mathbf{D}(\hat{\mathbf{x}})$  je menší než námi zvolený kontrastový práh, kandidát na klíčový bod je zamítnut.

Autor algoritmu SIFT doporučuje hodnotu kontrastového prahu 0.03[1], v knihovně OpenCV je defaultní hodnota 0.04 (v obou případech se uvažuje obor hodnot obrazové funkce jako interval  $\langle 0, 1 \rangle$ ).

Kromě bodů s nízkým kontrastem chceme také odfiltrovat body ležící na hranách, protože u těchto bodů je určení polohy mezi dvěma snímky nejednoznačné.

Tohoto dosáhneme stejnou technikou využívanou v Harrisově detektoru rohů [7].

Uvažujme vrstvu DoG space, ve které se nachází prověřovaný kandidát, jako povrch. Z křivosti tohoto povrchu určíme, jestli se bod nachází v rohu, na okraji nebo v rovině.

Výpočet využívá Hessovu matici ve tvaru

$$\mathbf{H} = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix} \quad (1.4)$$

kde  $D_{xx}, D_{yy}, D_{xy}$  jsou druhé derivace  $D$ . Podmínka pro rohový bod vychází z vlastností poměru charakteristických hodnot Hessovy matice a můžeme ji vyjádřit jako

$$\frac{\text{Tr}(\mathbf{H})}{\text{Det}(\mathbf{H})} < \frac{(r+1)^2}{r} \quad (1.5)$$

$$\text{Tr}(\mathbf{H})^2 = D_{xx} + D_{yy} \quad (1.6)$$

$$\text{Det}(\mathbf{H}) = D_{xx}D_{yy} - (D_{xy})^2 \quad (1.7)$$

kde  $r$  je námi zvolený parametr, autor algoritmu doporučuje hodnotu 10[1].

## Přidělení orientace

Abychom dosáhli rotační invariance deskriptoru, musíme každému klíčovému bodu přidělit údaj o jeho orientaci. Tuto orientaci určíme analýzou okolí klíkového bodu o stejné velikosti, jako byla velikost příslušného Gaussova filtru.

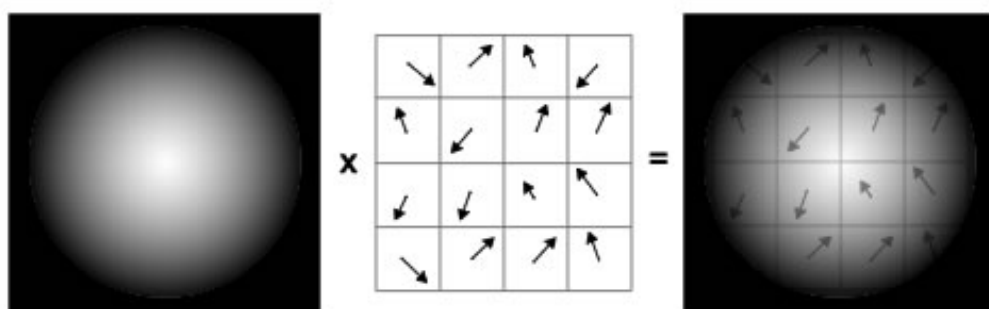
Z tohoto okna vypočítáme gradient a v každém bodě určíme jeho velikost a úhel. Funkce velikosti gradientu:

$$m(x, y) = \sqrt{(L(x+1, y) - L(x-1, y))^2 + (L(x, y+1) - L(x, y-1))^2} \quad (1.8)$$

funkce úhlu gradientu:

$$\theta(x, y) = \arctan((L(x+1, y) - L(x-1, y)) / (L(x, y+1) - L(x, y-1))) \quad (1.9)$$

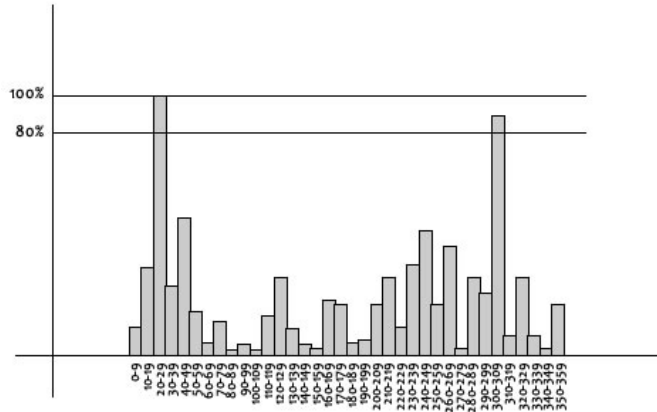
Vytvoříme vážený histogram hodnot  $\theta(x, y)$  s 36 intervaly, jako váhu každé hodnoty použijeme součin  $m(x, y)$  a hodnoty Gaussova filtru v  $(x, y)$ .



Obr. 1.5: Vážení orientací gradientu gaussovým filtrem

Orientace klíkového bodu je nejčetnější úhel v histogramu.

Pokud se v histogramu nachází další úhel, jehož četnost je alespoň 80% nejvyšší četnosti, vytvoříme s touto orientací nový klíčový bod na stejných souřadnicích.

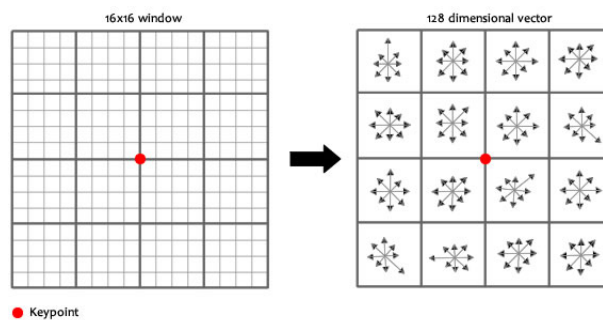


Obr. 1.6: Histogram hodnot  $\theta(x, y)$

## Deskriptor klíčového bodu

Pro vybudování deskriptoru vytvoříme okno  $16 \times 16$  hodnot kolem klíčového bodu. Jsou to mezipixelové hodnoty získané bilineární interpolací. V tomto okně vypočítáme velikost a orientaci gradientu podobně jako u přidělování orientace, s tím rozdílem, že od hodnot orientace gradientu odečteme hodnotu orientace klíčového bodu zjištěnou v předchozím kroku. Tím dosáhneme rotační invariance deskriptoru.

Tentokrát nebudeme počítat jeden histogram s 36 intervaly, ale  $4 \times 4$  histogramy každý s osmi intervaly. Každému histogramu tedy bude odpovídat podokno  $4 \times 4$  hodnot.



Obr. 1.7: Ilustrace způsobu získání deskriptoru

Hodnoty v histogramu normalizujeme, čímž zmenšíme vliv světelných podmínek při snímání. Deskriptor obsahuje  $4 \times 4 \times 8 = 128$  hodnot.

### 1.1.3 SURF - Speeded up robust features

Algoritmus SURF[2] má stejný cíl jako SIFT - tedy nalezení a popis kvalitních klíčových bodů. SURF je ale modernější algoritmus, který vznikl částečně jako reakce na SIFT a snaží se jej překonat v robustnosti, rozlišitelnosti a hlavně rychlosti výpočtu.[2] Vyšší rychlosti dosahuje hlavně díky použití jednoduchých aproximací namísto složitých výpočtů.

#### Detekce klíčových bodů

Jak již bylo zmíněno, algoritmus pro vyhledávání klíčových bodů musí být nezávislý na zvětšení. SIFT tohoto dosahuje analýzou maxim a minim obrazů Difference of Gaussians (které slouží jako aproximace výpočetně náročnějšího Laplacian of Gaussian). SURF využívá výsledek konvoluce s druhými derivacemi Gaussiánů, které ve skutečnosti aproximuje prostými obdelníkovými filtry.

U SIFTu se k detekci kvalitních klíčových bodů využívá pyramidu Gaussiánů o postupně zvětšujícím se parametru  $\sigma$  a postupným zmenšením podvzorkováním. SURF dosahuje podobného efektu postupným zvětšením filtru.

Postupnou aplikací zvětšujících se filtrů se vytváří scale space. Klíčové body jsou takové body ve scale space, které mají maximální determinant Hessovy matice v daném okolí dvaceti sedmi bodů (podobně jako v Obr. 1.3).

Hessova matice má tvar

$$\mathbf{H}(x, y, \sigma) = \begin{bmatrix} L_{xx}(x, y, \sigma) & L_{xy}(x, y, \sigma) \\ L_{xy}(x, y, \sigma) & L_{yy}(x, y, \sigma) \end{bmatrix} \quad (1.10)$$

kde

$$L_{xx}(x, y, \sigma) = I(x, y) * \frac{\partial^2}{\partial x^2} g(\sigma) \quad (1.11)$$

$$L_{xy}(x, y, \sigma) = I(x, y) * \frac{\partial^2}{\partial xy} g(\sigma) \quad (1.12)$$

a  $g(\sigma)$  je jádro Gaussova filtru.

Právě členy  $L_{xx}$ ,  $L_{xy}$  a  $L_{yy}$  jsou velmi náročné z hlediska výpočtu.

Druhá derivace Gaussova jádra se nahrazuje jednoduchými obdelníkovými jádry (Obr. 1.8).

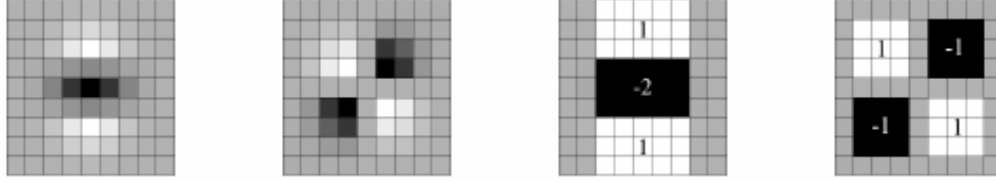
Tyto náhrady  $L_{xx}$ ,  $L_{xy}$  a  $L_{yy}$  se označují  $D_{xx}$ ,  $D_{xy}$  a  $D_{yy}$ .

Aproximovaný determinant se vypočte

$$\text{Det}(\mathbf{H}) = D_{xx}D_{yy} - (wD_{xy})^2 \quad (1.13)$$

Parametr  $w$  je nezbytný, aby byla zachována ekvivalentní energie mezi pravými Gaussovými jádry a jádry aproximovanými.

Přestože je hodnota  $w$  teoreticky závislá na  $\sigma$  nahrazovaného filtru a velikosti obdelníkové náhrady, prakticky může být hodnota konstantní,  $w = 0.9$ .



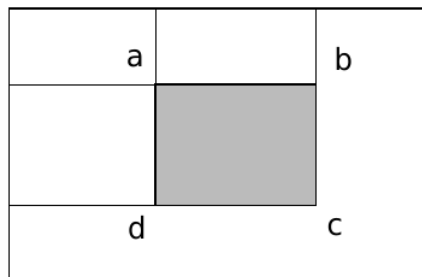
Obr. 1.8: (vlevo) Diskrétní druhá derivace Gaussiánu ve směrech  $yy$  a  $xy$  a jejich aproximace (vpravo)

Konvoluce s použitím těchto filtrů může být velmi efektivně vypočítána pomocí *integrálních obrazů*. Integrální obraz  $I_\Sigma$  v bodě  $(x, y)$  reprezentuje součet všech hodnot obrazové funkce  $I$  původního snímku v obdelníkovém okně mezi bodem  $(0, 0)$  a bodem  $(x, y)$ .

$$I_\Sigma(x, y) = \sum_{i=0}^{x} \sum_{j=0}^{y} I(i, j) \quad (1.14)$$

Z integrálního obrazu jednoduše zjistíme součet hodnot jakékoliv podoblasti.

$$\Sigma_{abcd} = I_\Sigma(x_c, y_c) - I_\Sigma(x_b, y_b) - I_\Sigma(x_d, y_d) + I_\Sigma(x_a, y_a) \quad (1.15)$$

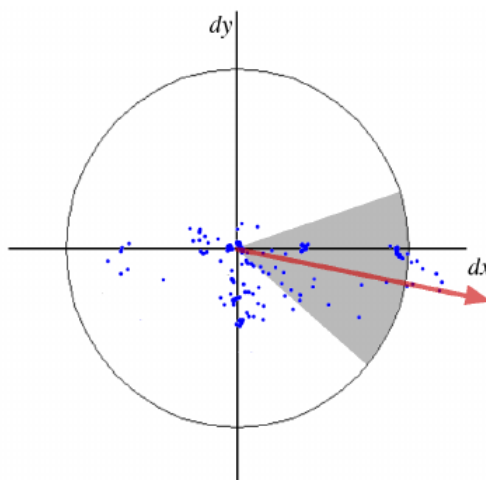


Obr. 1.9: Podoblast integrálního obrazu ohraničená body  $a, b, c, d$

Tento součet vydělíme počtem pixelů v oblasti a získáme výsledek obdelníkového průměrovacího filtru.

## Přidělení orientace

Pro určení orientace klíčového bodu se vypočítají waveletové odezvy (Haarovy wavelety) v kruhovém okolí klíčového bodu. Potom výsledný kruhový obraz postupně prochází rotující kruhovou výsečí s úhlem  $60^\circ$ . Orientace klíčového bodu určuje úhel nastavení, pro který je součet obsažených hodnot ve výseči maximální.



Obr. 1.10: Kruhová výseč určující orientaci klíčového bodu SURF

## Deskriptor klíčového bodu

Deskriptor je vytvořený z hodnot ve čtvercovém okolí klíčového bodu. Tento čtverec je orientovaný podle orientace klíčového bodu určené v předchozím kroku algoritmu a je rozdělen na  $4 \times 4$  menších čtverců. V těchto malých čtvercích jsou vypočteny waveletové odezvy ve směrech  $x$  a  $y$  (značí se  $dx$  a  $dy$ ) relativně k orientaci klíčového bodu. Z každého malého čtverce je určen vektor

$$\mathbf{v} = \left( \sum dx, \sum |dx|, \sum dy, \sum |dy| \right) \quad (1.16)$$

Deskriptor klíčového bodu je tvořen vektory  $\mathbf{v}$  každého malého podčtverce, celkem tedy  $4 \times 4 \times 4 = 64$  hodnot.



## 1.2 Korespondence klíčových bodů

Nalezení korespondujících dvojic obrazových bodů odpovídajících stejnému prostoro-  
rovému bodu ve dvou snímcích probíhá srovnáváním deskriptorů mezi množinami  
bodů z prvního a druhého snímku. Vzhledem k tomu, že deskriptory byly získány  
z různých snímků, musíme uvažovat, že deskriptor bodu z každého snímku bude  
mírně odlišný. Nemůžeme proto hovořit o absolutní shodě deskriptorů. Namísto toho  
musíme nějakým způsobem určit míru jejich podobnosti.

Uvažujme o deskriptoru klíčového bodu jako vektoru  $n$  hodnot (např. pro algo-  
ritmus SIFT platí  $n = 128$ ), potom podobnost deskriptorů vyhodnotíme výpočtem  
Minkowského vzdálenosti neboli  $L_p$  metriky mezi různými kandidáty v množinách  
klíčových bodů. Jako korespondenci určíme dvojici s nejnižší Minkowského vzdále-  
ností.

**Minkowského vzdálenost** ( $L_p$  metrika) se obecně pro dané  $p$  vypočte jako

$$d^{L_p}(\mathbf{x}, \mathbf{x}') = \left( \sum_{i=1}^n |x_i - x'_i|^p \right)^{\frac{1}{p}} \quad (1.17)$$

Nejčastěji používané hodnoty  $p$  jsou  $p = 1$  a  $p = 2$

Pro  $p = 1$  (Cityblock/Manhattan/Taxicab distance) platí:

$$d^{L_1}(\mathbf{x}, \mathbf{x}') = \sum_{i=1}^n |x_i - x'_i| \quad (1.18)$$

a pro  $p = 2$  (Euclidean distance)

$$d^{L_2}(\mathbf{x}, \mathbf{x}') = \sqrt{\sum_{i=1}^n (x_i - x'_i)^2} \quad (1.19)$$

Metrika  $L_1$  bude rychlejší z hlediska výpočtu, protože nevyužívá operace jako  
umocnění a odmocnění. Na druhou stranu  $L_2$  je bližší geometrickému významu  
velikosti vektoru rozdílů.

Při určování bodových korespondencí je nutné také uvažovat různé důvody  
pro falešnou podobu nebo nepodobu deskriptorů. Mezi takové důvody patří například:

- Podobné oblasti v obrázku (opakující se vzor)
- Deformace povrchu pod různými snímacími úhly
- Zastínění objektu
- Průhledné nebo reflexní povrchy

### 1.2.1 Detekce falešných korespondencí pomocí poměru Minkowského vzdáleností

Tato metoda pracuje kromě vzdálenosti nejlepší shody  $m = \min d^{L_p}(\mathbf{x}, \mathbf{x}')$  deskriptoru klíčového bodu také se vzdáleností druhé nejlepší shody  $n = \min_{\text{kromě } m} d^{L_p}(\mathbf{x}, \mathbf{x}')$ .

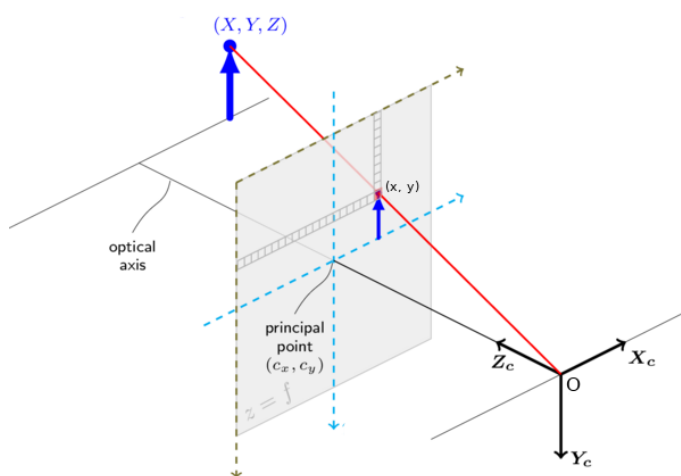
Pokud poměr je  $\frac{m}{n}$  příliš vysoký (autor metody R. Lowe doporučuje práh poměru  $\frac{m}{n} > 0.8$ ) uvažujeme, že  $m$  i  $n$  reprezentují možnou korespondenci. Jelikož nejsme schopni rozhodnout, která korespondence je ta správná, tento klíčový bod úplně vyřadíme.

## 1.3 Prostorová rekonstrukce scény

### 1.3.1 Model kamery

Pro matematické vyjádření vlastností kamery se využívá model *pinhole camera* (také *camera obscura*). V tomto modelu je kamera reprezentována čtyřbokým jehlanem, jehož podstavu tvoří snímáný obraz a špičku ohnisko.

Výšku jehlanu určuje *ohnisková vzdálenost*.



Obr. 1.11: Pinhole camera

Při zpracování reálných snímků je nutné také uvažovat o zkreslení obrazu optickou soustavou použité kamery. Toto zkreslení má za následek nesprávné polohy obrazových bodů z hlediska pinhole modelu, což by vedlo k nesprávné interpretaci klíčových bodů. Radiální zkreslení je způsobeno tvarem použité čočky, tangenciální je způsobeno nedokonalým zarovnáním optické osy čočky a normály senzoru. Geometrické zkreslení obrazu lze charakterizovat distorčními koeficienty  $(k_1, k_2, k_3, p_1, p_2)$ .

Opravené obrazové souřadnice pro radiální zkreslení se vypočtou:

$$x' = x(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \quad (1.20)$$

$$y' = y(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \quad (1.21)$$

Pro tangenciální zkreslení:

$$x' = x + [2p_1 xy + p_2(r^2 + 2x^2)] \quad (1.22)$$

$$y' = y + [p_1(r^2 + 2y^2) + 2p_2 xy] \quad (1.23)$$

Parametr  $r$  určuje vzdálenost bodu  $(x, y)$  od středu obrazu.

Jakýkoli viditelný prostorový bod se promítne z prostorových souřadnic  $(X, Y, Z)$  do obrazových souřadnic  $(x, y)$  přes paprsek, který směřuje do ohniska. Pozice kamery v prostoru je udávána pozicí ohniska a orientace je udávána orientací optické osy (polopřímka z ohniska do středu snímaného obrazu). Toto promítnutí charakterizuje tzv. *Projekční matice*.

### Projekční matice

Projekční matice (*Projection matrix, Camera matrix*)  $P$  charakterizuje promítnutí bodu z homogenních prostorových souřadnic  $(X, Y, Z, 1)^T$  do homogenních obrazových souřadnic  $(x, y, 1)$ :

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \mathbf{P}_{3 \times 4} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (1.24)$$

$$\mathbf{x} = \mathbf{P}\mathbf{X} \quad (1.25)$$

Projekční matice obsahuje jak údaje o vlastnostech snímacího systému (intrinsické parametry), tak o jeho poloze a orientaci v 3D rámci (extrinsické parametry).

$$\mathbf{P} = \mathbf{K}[\mathbf{R}|\mathbf{t}] \quad (1.26)$$

kde  $K$  je  $3 \times 3$  matice intrinsických parametrů,  $\mathbf{R}$  je  $3 \times 3$  matice rotace a  $\mathbf{t}$  je  $3 \times 1$  vektor translace. Analýzou dílčích parametrů můžeme určit, že projekční matice má celkem jedenáct stupňů volnosti.[15]

**Intrinsické parametry** se vyjadřují ve formě matice  $\mathbf{K}$  (intrinsická matice/kalibrační matice):

$$\mathbf{K} = \begin{bmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

Parametry  $f_x, f_y, c_x, c_y$  jsou parametry vycházející z modelu kamery obskury.  $f_x$  a  $f_y$  označují ohniskovou vzdálenost.  $c_x, c_y$  jsou souřadnice principiálního bodu, tedy bodu, který se nachází v průsečíku optické osy kamery a snímku. Parametr  $s$  je zkosení (*skew*) a popisuje situaci, kdy osy  $x$  a  $y$  obrazové plochy nejsou kolmé. Ohniskové vzdálenosti v  $x$  a  $y$  budou v ideálním případě obě stejné, tedy  $f_x = f_y$ , ale při reálné kalibraci se mohou lišit. Důvodem k odlišnosti mohou být například chyby senzoru kamery, neuniformní zvětšení snímku nebo zkreslení způsobené čočkou kamery.

**Extrinsické parametry** kamery jsou údaje o jejím posunutí  $\mathbf{t}$  (vektor translace) a jejím natočení  $\mathbf{R}$  (matice rotace). Přestože má matice rotace pouze tři stupně volnosti (rotace kolem každé z os v trojrozměrném prostoru), se používá ve tvaru  $3 \times 3$ . (Reprezentaci rotace se věnuje příloha C.) Samotnou polohu kamery  $\mathbf{O}$  ve scéně určíme:

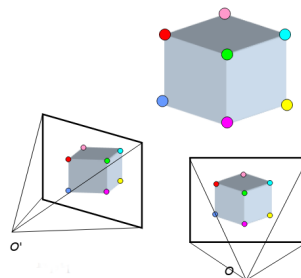
$$\mathbf{R}\mathbf{O} + \mathbf{t} = \mathbf{0} \quad (1.27)$$

z toho vyjádříme  $\mathbf{O}$

$$\mathbf{O} = -\mathbf{R}^T \mathbf{t} \quad (1.28)$$

### 1.3.2 Model dvou pohledů

Modelem dvou pohledů se myslí situace, kde trojrozměrná scéna je zachycená ve dvou různých snímcích zachycených pinhole kamerami. Vzájemná poloha a orientace těchto dvou kamer je definována třemi translačními a třemi rotačními parametry. Pokud jsou známy intrinsické a extrinsické parametry kamer, hovoříme o kalibrovaném případě.



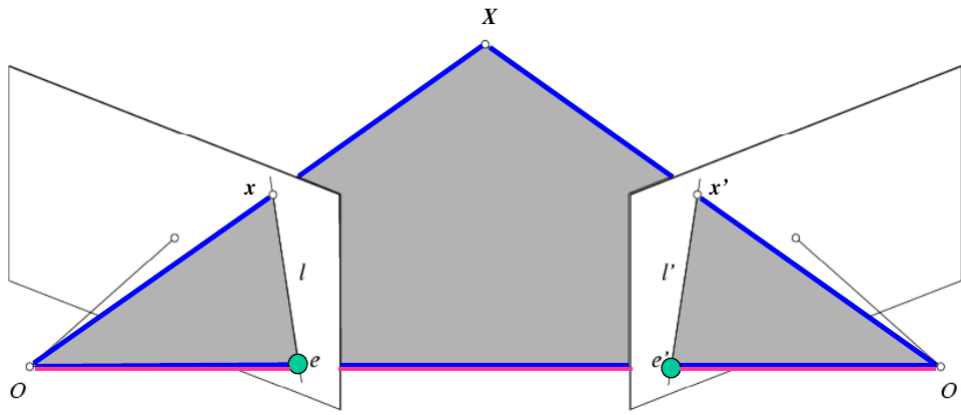
Obr. 1.12: Model dvou pohledů

## Epipolární geometrie

V epipolární geometrii se pracuje se vztahy a údaji, které vycházejí z modelu dvou pohledů, a lze ji využít k odhadnutí vzájemné polohy a orientace kamer z pouhé znalosti bodových korespondencí. Základním vztahem v epipolární geometrii je *koplanární omezení* (*Coplanarity constraint*). Koplanární omezení říká, že vektory spojující středy kamer se společným prostorovým bodem ( $\mathbf{OX}$ ,  $\mathbf{O'X}$ ) a vektor spojující středy kamer ( $\mathbf{OO'}$ ), leží ve stejné rovině.

Vzorec koplanárního omezení (vychází ze smíšeného součinu pro výpočet objemu rovnoběžnostěnu):

$$|\mathbf{O'X} \cdot (\mathbf{OO'} \times \mathbf{OX})| = 0 \quad (1.29)$$



Obr. 1.13: Epipolární geometrie

Obrazové souřadnice odpovídající vektorům  $\mathbf{OX}$  a  $\mathbf{O'X}$  můžeme vyjádřit jako  $\mathbf{x}$  a  $\mathbf{x'}$ , zvolíme-li  $\mathbf{O}$  jako počátek souřadnicové soustavy, tedy

$$\mathbf{x} = \mathbf{PX} = \mathbf{K}[\mathbf{I} \mid \mathbf{0}]\mathbf{X} \quad (1.30)$$

$$\mathbf{x'} = \mathbf{P'X} = \mathbf{K'}[\mathbf{R} \mid \mathbf{t}]\mathbf{X} \quad (1.31)$$

Translaci vyjádříme ve formě antisymetrické matice:

$$[\mathbf{t}]_{\times} = \begin{bmatrix} 0 & -t_3 & t_2 \\ t_3 & 0 & -t_1 \\ -t_2 & t_1 & 0 \end{bmatrix} \quad (1.32)$$

Koplanární omezení (1.29) můžeme potom vyjádřit i jako:

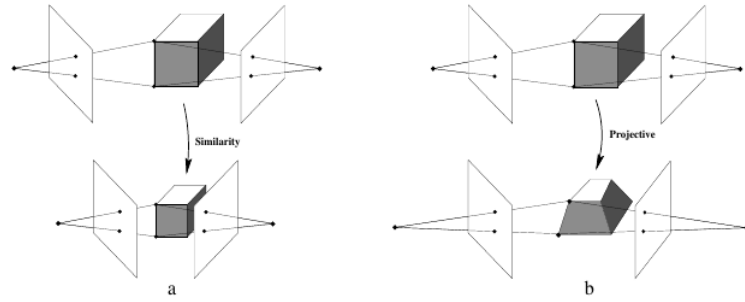
$$\mathbf{x'^T} \mathbf{K'^{-T}} [\mathbf{t}]_{\times} \mathbf{R} \mathbf{K}^{-1} \mathbf{x} = 0 \quad (1.33)$$

$$\mathbf{x'^T} \mathbf{F} \mathbf{x} = 0 \quad (1.34)$$

Toto platí pro každý pár korespondujících bodů ( $\mathbf{x'}$ ,  $\mathbf{x}$ ).  $\mathbf{F}$  se nazývá fundamentální matice (*fundamental matrix*). [8]

**Fundamentální matice** má sedm stupňů volnosti a obsahuje veškeré informace o relativní poloze a orientaci páru kamer, které lze získat ze znalosti párů korespondujících bodů, a bez dalších informací o scéně. Ze samotné fundamentální matice není možné sestavit euklidovskou (metrickou) rekonstrukci prostorových bodů - je možná maximálně projektivní rekonstrukce, tj. topologie rekonstruovaného objektu je správná, ale vzdálenosti a úhly nemusí být. (Obr.1.14(b))

Pro vytvoření metrické (euklidovské) rekonstrukce je třeba převést fundamentální matici do normalizovaných obrazových souřadnic. Takováto matice se nazývá esenciální matice (*essential matrix*).



Obr. 1.14: (a)nejednoznačnost měřítka, (b)projektivní nejednoznačnost [8]

**Esenciální matice** je fundamentální matice, ze které byly odstraněny vlivy intrinsických parametrů:

$$\mathbf{E} = \mathbf{K}'^T \mathbf{F} \mathbf{K} \quad (1.35)$$

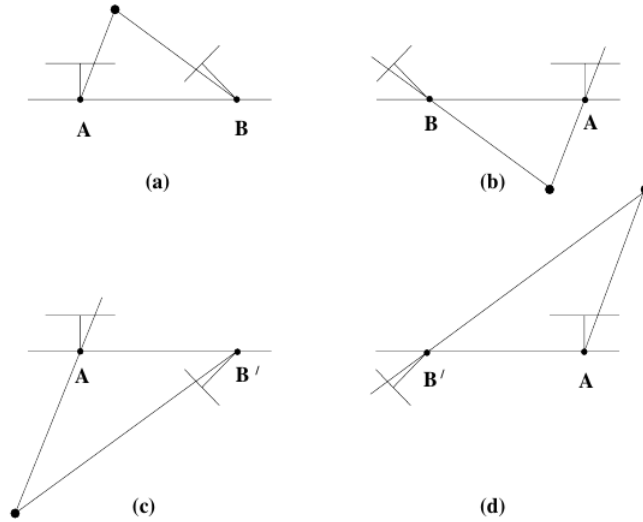
$$\mathbf{E} = \mathbf{K}'^T \mathbf{K}'^{-T} [\mathbf{t}]_{\times} \mathbf{R} \mathbf{K}^{-1} \mathbf{K} \quad (1.36)$$

$$\mathbf{E} = [\mathbf{t}]_{\times} \mathbf{R} \quad (1.37)$$

Pomocí esenciální matice lze získat rekonstrukci velikostí úhlů a relativních délek, stále však není možné získat absolutní měřítko scény. (Obr.1.14(a), Obr. 1.14(b))

### Extrakce projekčních matic z esenciální matice

Zavedeme první projekční matici  $\mathbf{P} = [\mathbf{I} \mid \mathbf{0}]$ , potom  $\mathbf{P}' = [\mathbf{R} \mid \mathbf{t}]$ . Pomocí singulárního rozkladu matice  $\mathbf{E}$  získáme čtyři možná řešení pro dvojce  $\mathbf{R}$  a  $\mathbf{t}$ . (Obr. 1.15)



Obr. 1.15: Možné kombinace projekčních matic získaných singulárním rozkladem esenciální matice [8]

### Výpočet Fundamentální matice

**Osmibodová metoda** je nejjednodušším způsobem výpočtu fundamentální matice z bodových korespondencí. Název napovídá, že pro jeho použití je nezbytné znát alespoň osm párů korespondujících bodů  $(\mathbf{x}'_i, \mathbf{x}_i); i = 1, \dots, n; n \geq 8$ .

Požadavek na osm bodů je zdůvodněn homogenní povahou fundamentální matice. Homogenní matice  $3 \times 3$  má obecně 8 stupňů volnosti.

Princip osmibodového algoritmu vychází se z koplanárního omezení(1.34):

$$\mathbf{x}'^T \mathbf{F} \mathbf{x} = 0 \quad (1.38)$$

$$\begin{bmatrix} x & y & 1 \end{bmatrix} \cdot \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = 0 \quad (1.39)$$

$$xx'f_{11} + xy'f_{21} + xf_{31} + yx'f_{12} + yy'f_{22} + yf_{32} + x'f_{13} + y'f_{23} + f_{33} = 0 \quad (1.40)$$

To samé musí platit pro všechny páry  $(\mathbf{x}', \mathbf{x})$ , tudíž:

$$\underbrace{\begin{bmatrix} x_1x'_1 & x_1y'_1 & x_1 & y_1x'_1 & y_1y'_1 & y_1 & x'_1 & y'_1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_nx'_n & x_ny'_n & x_n & y_nx'_n & y_ny'_n & y_n & x'_n & y'_n & 1 \end{bmatrix}}_A \cdot \underbrace{\begin{bmatrix} f_{11} \\ f_{21} \\ f_{31} \\ f_{12} \\ f_{22} \\ f_{32} \\ f_{13} \\ f_{23} \\ f_{33} \end{bmatrix}}_{\mathbf{f}} = 0 \quad (1.41)$$

$$A\mathbf{f} = 0 \quad (1.42)$$

Výraz  $A\mathbf{f} = 0$  lze řešit singulárním rozkladem

$$A = UDV^T \quad (1.43)$$

Potom  $\hat{\mathbf{f}}$  je poslední sloupec matice  $V$ . Matice  $\hat{\mathbf{F}}$  odpovídající vektoru  $\hat{\mathbf{f}}$  však nemá zaručenou hodnotu 2. Pro zaručení podmínky hodnoty se provede druhá singulární dekompozice:

$$\hat{\mathbf{F}} = U'D'V'^T \quad (1.44)$$

Nová matice  $\mathbf{F}$  vznikne rekompozicí se zaměněnou diagonální maticí  $D$ :

$$\mathbf{F} = U' \begin{bmatrix} d'_{11} & 0 & 0 \\ 0 & d'_{22} & 0 \\ 0 & 0 & 0 \end{bmatrix} V'^T \quad (1.45)$$

$\mathbf{F}$  má sedm stupňů volnosti, z čehož vyplývá možnost výpočtu za pomoci sedmi párů bodů. Taková metoda existuje, ale prakticky se nepoužívá, protože je komplikovanější a protože klíčových bodů není nedostatek.

**RANSAC metoda** (RANdom SAmple Consensus) je obecně metoda hledání modelu dat, která jsou zatížena výrazným šumem. V kontextu výpočtu fundamentální matice řeší problém vlivu falešných korespondencí. Algoritmus pracuje následujícím způsobem:

1. Pseudonáhodný výběr minimálního počtu vzorků nezbytného pro výpočet fundamentální matice (8)
2. Výpočet kandidáta fundamentální matice  $\mathbf{F}_c$  osmibodovým algoritmem z vybraných vzorků
3. Zjištění počtu všech párů, které souhlasí s kandidátem fundamentální matice (inliers) a těch které nesouhlasí (outliers)



4. Opakování kroků 1 → 3 pro různé volby vzorků
5. Výběr kandidáta s největším počtem souhlasných párů (inliers)

Úspěšnost RANSACu závisí na volbě strategie výběru vzorkových množin a na volbě podmínky definující inliery a outliery. Taková podmínka vychází z definice fundamentální matice (1.34). Pokud pár bodů  $(\mathbf{x}'_i, \mathbf{x}_i)$  nesplňuje s kandidátem tuto definici, je označen za outliera.

V reálných datech musíme uvažovat, že nedokonalosti dat způsobené například nedokonalým odstraněním zkreslení nebo nepřesným určením obrazových souřadnic klíčových bodů způsobí, že podmínku  $\mathbf{x}'_i{}^T \mathbf{F}_c \mathbf{x}_i = 0$  nebude zřejmě dokonale splňovat žádný pár bodů. Podmínka je tedy založená na rozhodnutí, jak velkou odchylku od ideální rovnosti jsme schopni tolerovat.

RANSAC oproti jednoduchému osmibodovému algoritmu přináší nejen robustnější výsledky, ale automaticky i informaci o tom, které páry bodů nesouhlasí s tímto výsledkem - o takových výsledcích můžeme z jistotou prohlásit, že se jedná o falešné korespondence a z dalšího uvažování je vyřadit.

## Triangulace

Triangulace je výpočet 3D souřadnic prostorového bodu  $\mathbf{X} = (X, Y, Z)^T$  ze známých obrazových souřadnic jeho průmětů  $\mathbf{x} = (x, y)^T$  a  $\mathbf{x}' = (x', y')^T$  podle projekčních matic  $\mathbf{P}$  a  $\mathbf{P}'$ .

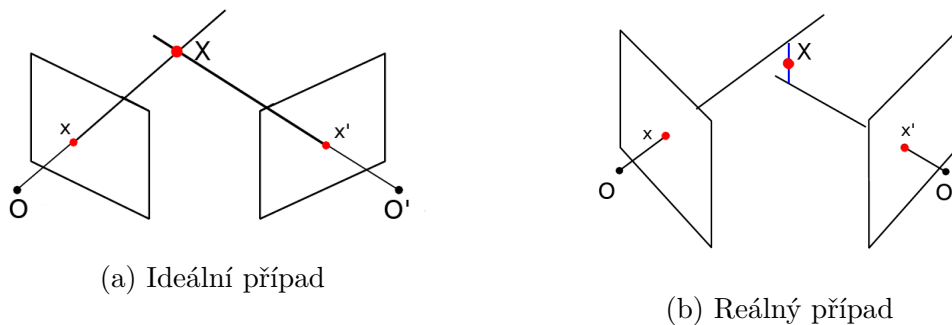
Projekční matice mají tvar:

$$\mathbf{P} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix} = \begin{bmatrix} \mathbf{p}^{1T} \\ \mathbf{p}^{2T} \\ \mathbf{p}^{3T} \end{bmatrix} \quad \mathbf{P}' = \begin{bmatrix} \mathbf{p}'^{1T} \\ \mathbf{p}'^{2T} \\ \mathbf{p}'^{3T} \end{bmatrix}$$

( $\mathbf{p}^{iT}$  jsou řádky matice  $\mathbf{P}$ ) Prostorové souřadnice  $\mathbf{X}$  získáme z řešení vztahu

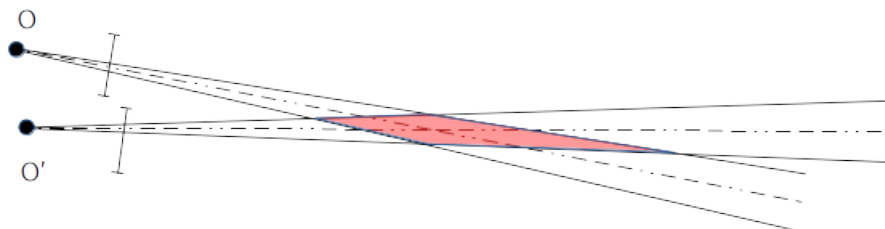
$$\begin{bmatrix} x\mathbf{p}^{3T} - \mathbf{p}^{1T} \\ y\mathbf{p}^{3T} - \mathbf{p}^{2T} \\ x'\mathbf{p}'^{3T} - \mathbf{p}'^{1T} \\ y'\mathbf{p}'^{3T} - \mathbf{p}'^{2T} \end{bmatrix} \mathbf{X} = 0 \quad (1.46)$$

Je nutné si uvědomit, že v reálném případě se parametry projekčních matic nepodaří přesně určit a proto průsečík paprsků nebude nezbytně existovat. Triangulace se tedy promění v úlohu hledání bodu blízkého dvou přímkám (Obr. 1.16).



Obr. 1.16: Ideální a reálný případ triangulace

Dále je vhodné uvažovat, že i malé chyby v určení projekční matice budou mít za následek velkou chybu v prostorových souřadnicích, hlavně ve směru projekčních paprsků (Obr. 1.17).



Obr. 1.17: Možné výsledky triangulace při malých chybách určení projekčních paprsků

### 1.3.3 Postupné skládání mnoha pohledů

Pomocí vztahů popsanych v předchozích kapitolách je nyní možné sestavit jednoduchý algoritmus pro rekonstrukci trojrozměrných souřadnic bodů snímané scény a poloh a orientací obou kamer:

1. Odstranění zkreslení - 1.3.1
2. Nalezení klíčových bodů a příslušných deskriptorů - 1.1.1
3. Nalezení korespondencí klíčových bodů - 1.2
4. Výpočet esenciální matice - 1.3.2
5. Konstrukce projekčních matic z esenciální matice - 1.3.2
6. Triangulace klíčových bodů do prostoru - 1.3.2

Zdalo by se, že opakovanou aplikací algoritmu pro dva pohledy na seznam po sobě jdoucích snímků (např. z videozáznamu), by mělo být možné získat rekonstrukci trajektorie kamery a také podrobnou rekonstrukci scény z mnoha úhlů. Z čistě

matematického hlediska tomuto postupu nic nebrání. Při praktické implementaci však narazíme na řadu problémů, veskrze způsobených nepřesnostmi dílčích výpočtů, které zabrání použití takto jednoduché metody. Možné řešení těchto problémů je popsáno v 1.3.4 - Bundle Adjustment.

### Problémy postupného skládání

**Nesouhlasná triangulace** je případ, kdy triangulace stejného bodu z různých dvojic pohledů vyprodukuje různé prostorové souřadnice. Hlavním původcem takového problému je nepřesné určení vzájemné polohy a orientace pohledů před triangulací. Přestože je tento problém velmi výrazný při jednoduchém skládání, princip podobný nesouhlasnému promítnutí se používá jako součást pokročilejší metody, kde napomáhá k zpřesnění parametrů rekonstrukce (více v 1.3.4).

**Chyby triangulace** jsou již popsány v 1.3.2. Původcem mohou být jak nepřesnosti v určení projekčních matic - řešeno v 1.3.4, tak falešné korespondence klíčových bodů - 1.2.

**Drift** je chyba určení polohy a orientace kamery způsobená chybami ze všech předchozích určení polohy a orientace na trajektorii kamery.

### 1.3.4 Matematická formulace pro $N$ bodů a $M$ pohledů

Předchozí kapitoly 1.3.2 - Model dvou pohledů a 1.3.3 - Postupné skládání mnoha pohledů se zabývaly rekonstrukcí ze dvou sousedních pohledů. Pro napravení problémů způsobených tímto způsobem je vhodné uvažovat vztahy mezi mnoha pozorováními stejných bodů najednou.

Matematicky lze projekci  $N$  bodů a  $M$  pohledů formulovat následovně:

$$\mathbf{x}_{ij} = \mathbf{P}_j \mathbf{X}_i \quad (1.47)$$

$$i = 1, 2, \dots, N, \quad j = 1, 2, \dots, M \quad (1.48)$$

*Záznam klíčového bodu  $ij$  je projekcí  $i$ -tého prostorového bodu v  $j$ -tém pohledu.*

Z toho vychází nový přístup k rekonstrukci scény z mnoha pohledů:

Hledá se taková kombinace parametrů projekčních matic  $\mathbf{P}$  a prostorových souřadnic  $\mathbf{X}$ , aby odpovídaly příslušným naměřeným obrazovým souřadnicím  $\mathbf{x}$ .

### Optimalizační úloha - Bundle Adjustment

Uvažujme nyní o problému rekonstrukce jako o nelineární optimalizační úloze založené na vztahu 1.47. Neznámými budou projekční matice  $\mathbf{P}$  a prostorové souřadnice

bodů  $\mathbf{X}$ . Jako optimalizační kritérium zvolíme odchylku naměřených obrazových souřadnic  $\mathbf{x}$  od reprojekčních obrazových souřadnic  $\hat{\mathbf{x}} = \mathbf{P}\mathbf{X}$ .

$$\min_{\mathbf{P}_j, \mathbf{X}_i} \sum_{i=1}^N \sum_{j=1}^M d(\mathbf{x}_{ij}, \mathbf{P}_j \mathbf{X}_i) \quad (1.49)$$

$$d(\mathbf{x}, \hat{\mathbf{x}}) = d^{L_2}(\mathbf{x}, \hat{\mathbf{x}}) = \sqrt{(x - \hat{x})^2 + (y - \hat{y})^2} \quad (1.50)$$

Analýzou počtu a stupňů volnosti neznámých můžeme popsat složitost výpočtu problému:

- N bodů
- M pohledů
- intrinsické parametry  $\mathbf{P}$  - 5 stupňů volnosti
- extrinsické parametry  $\mathbf{P}$  - 6 stupňů volnosti
- $\mathbf{X}$  - 3 stupně volnosti

Celkový počet neznámých je  $6M + 3N$ , uvažujeme-li, že intrinsické parametry jsou předem známé a pro každý pohled stejné.

Počet parametrů kritéria bude zdánlivě  $M \cdot N$ . Tato hodnota však uvažuje, že každý prostorový bod se zobrazí do každého snímku, v realitě tomu tak ale nebude.

Uvažujme o vztahu 1.47 jako o transformaci z prostoru neznámých  $[\mathbf{P}_1, \dots, \mathbf{P}_M, \mathbf{X}_1, \dots, \mathbf{X}_N]$  do prostoru známých bodů  $[\mathbf{x}_1, \dots, \mathbf{x}_{N \cdot M}]$ . Jakobián této transformace bude mít formu řídké matice, protože každý element kritéria je ovlivněn pouze jednou projekční maticí a jedněmi prostorovými souřadnicemi.

Tento poznatek může spolu s vhodnou volbou optimalizační metody výrazně snížit výpočetní nároky.

Takovou metodou je například *Trust Region Reflective Algorithm*[21]. Ukázka řídké matice je v kapitole 2.3.2

## 2 PRAKTICKÁ ČÁST

### 2.1 Volba nástrojů

Při volbě programovacího jazyka pro implementaci programu byla hlavním kritériem zkušenost a má schopnost jazyk používat. Takto okamžitě nastávají tři možnosti: C++, MATLAB, Python. U všech těchto jazyků existují knihovny pro zpracování a vizualizaci obrazových dat, pro MATLAB je to Image Processing Toolbox a Computer Vision Toolbox, pro C++ a Python existuje knihovna OpenCV. Pro řešení jsem zvolil jazyk Python v kombinaci s knihovnami OpenCV a SciPy stack. Učinil jsem tak, protože s Pythonem umím ze zmíněných možností pracovat nejlépe, včetně některých knihoven třetích stran.

Python je vysokoúrovňový interpretovaný jazyk s dynamickými typy a zaměřením na objektově orientované programování. Z těchto vlastností vychází hlavní výhody oproti C++: umožňuje rapidní prototypování a práci v interaktivním režimu (podobně jako MATLAB).

Nástroj	Verze	Použití
Python	3.6.1	Interpreter + standardní knihovna
IPython	5.3.0	Pokročilé interaktivní prostředí
OpenCV	3.2.0	Vyhodnocování obrazových dat a epipolární geometrie
NumPy	1.12.1	Maticové datové typy a vektorizované operace
pandas	0.19.2	Indexované datové struktury, organizace a analýza dat
Matplotlib	2.0.0	Vizualizace
SciPy	0.19.0	Statistické, numerické a optimalizační funkce

Tab. 2.1: Verze a stručné popisy použitých knihoven

Knihovny *IPython*, *NumPy*, *pandas*, *Matplotlib* a *SciPy* jsou součástí open-source distribuce *SciPy Stack*, která tvoří kompletní sadu nástrojů pro vědecké programování, podobné MATLABu. Pythonové adaptéry pro OpenCV jsou kompatibilní s knihovnami *SciPy Stacku* a pro vstupy a výstupy používají maticové typy z NumPy.

## 2.2 Algoritmus rekonstrukce

V rámci praktické části práce jsem implementoval algoritmus pro rekonstrukci trojrozměrné scény ze záznamu kamery. Předpokládám, že intrinsické parametry a distorční koeficienty snímáče jsou známy předem.

Algoritmus pracuje následujícím způsobem:

### Inicializace

1. Načtení intrinsických parametrů a distorčních koeficientů  $\Rightarrow (\mathbf{K}, \mathbf{dk})$
2. Načtení snímků videozáznamu
3. Redukce počtu snímků záznamu
4. Odstranění zkreslení
5. Výběr prvního snímku  $\Rightarrow S_0$
6. Určení klíčových bodů a deskriptorů ve snímku  $S_0$
7. Zavedení projekční matice pro  $S_0$ :  $\mathbf{P}_0 = \mathbf{K}[\mathbf{I} \mid \mathbf{0}]$
8. Zavedení množiny trojrozměrných bodů (prozatím prázdná):  $\mathbf{X}$

### Cyklus $i = 1$

9. Výběr snímku:  $S_i$
10. Určení klíčových bodů a deskriptorů ve snímku  $S_i$
11. Vyhledání korespondujících párů bodů:  $(S_{i-1}, S_i) \Rightarrow \bar{C}$
12. Výpočet esenciální matice z korespondencí:  $(\bar{C}, \mathbf{K}) \Rightarrow (\mathbf{E}, \text{maska})$   
Odstranění párů které nesouhlasí s esenciální maticí:  $(\bar{C}, \text{maska}) \Rightarrow \hat{C}$
13. Extrakce rotace a translace z esenciální matice:  $\mathbf{E} \Rightarrow (\mathbf{R}, \mathbf{t})$
14. Zavedení projekční matice pro  $S_i$ :  $(\mathbf{K}, \mathbf{R}, \mathbf{t}, \mathbf{P}_{i-1}) \Rightarrow \mathbf{P}_i$
15. Přidání nových bodů triangulací:  $(\hat{C}, \mathbf{P}_{i-1}, \mathbf{P}_i, \mathbf{X}) \Rightarrow \mathbf{X}$
16.  $i = i + 1$
17. **Návrat do 9.**

### Ukončení

18. Bundle adjustment:  $([\mathbf{P}_0, \dots, \mathbf{P}_i], \mathbf{X}) \Rightarrow ([\mathbf{P}_0, \dots, \mathbf{P}_i], \mathbf{X})$
19. Statistická filtrace prostorových bodů
20. Zobrazení výsledků

## 2.3 Struktura programu

Rekonstrukční algoritmus je realizován třídou `Reconstruction`, která dědí ze třídy `BundleAdjuster`. Inicializace třídy `Reconstruction` bere dva argumenty: `detector` - typ detektoru klíčových bodů ("sift"/"surf") a `verbosity` (defaultní hodnota 0),

který udává výřečnost textového výstupu (0: žádný, 3:nejvyšší). Samotný výřečný výstup provádí metoda `vprint`.

Algoritmus rekonstrukce (2.2) se spustí voláním metody `Reconstruction.reconstruct()`. Tato metoda postupně cyklicky volá ostatní privátní metody.

Pro vizualizaci rekonstrukce slouží tyto metody:

Metoda	Popis
<code>plot_reprojections</code>	Snímky s promítnutými prostorovými body*
<code>plot_reprojection_error</code>	Reprojekční chyby ve snímcích*
<code>plot_scene</code>	Prostorová scéna, původní barvy bodů
<code>plot_scene_zscore</code>	Prostorová scéna, barvy bodů podle zscore
<code>plot_sparsity</code>	Řídká matice Bundle Adjustmentu

*Metody označené \* v defaultním chování vykreslují do souborů v adresáři `debug/`*

Tab. 2.2: Souhrn vizualizačních metod třídy `Reconstruction`

Příklad inicializace a spuštění je v souboru `run-reconstruction.py`.

### 2.3.1 Vstupní data

Program umožňuje načítání obrazových dat jak z videozáznamu, tak z jednotlivých snímků. Data jsou načtena metodami `Reconstruction.load_footage(footage_file, imat, dist_coefs)` pro načtení videozáznamu nebo `Reconstruction.load_directory(directory, imat, dist_coefs)` pro načtení snímků uložených v adresáři. Při volání obou metod je nutné dodat i argument popisující intrinsické parametry ve formátu matice (popsáno v 1.3.1), a argument distorčních koeficientů (pole v pořadí  $(k_1, k_2, p_1, p_2, k_3)$ ).

Obrazová data jsou spolu s informacemi o poloze a rotaci pohledu uchována v šedotónovém (pro hledání klíčových bodů) i barevném formátu (extrakce barev a atraktivnější výstup) ve třídě `View`.

Po načtení dat z videozáznamu je vhodné redukovat počet snímků. Příliš blízké snímky ponesou duplicitní informace a jejich zpracování nepřináší žádné zlepšení modelu a pouze zpomaluje výpočty a přidává příležitosti k nepřesnostem.

Redukci počtu snímků provádí metoda `Reconstruction.reduce_views(n)`, kde `n` udává výsledný počet snímků. Snímky jsou vzorkovány rovnoměrně z celé načtené sekvence.

Až je výsledný počet snímků znám, provede se korekce zkreslení(1.3.1) metodou `Reconstruction.undistort_images()`.

## 2.3.2 Přehled důležitých operací

### Registrace pozorování

Pro sjednocení měřítka mezi páry snímků a pro správnou funkci bundle adjustmentu je nutné sjednotit množiny trojrozměrných bodů pozorovaných v různých párech. Nejvýhodnější místo pro provedení této operace je ihned po zjištění korespondencí. Registrace pozorování vypadá takto:

```
# view1 = již známý pohled
# view2 = nový pohled
# good = seznam korespondencí mezi view1 a view2
# self.observations = datový rámec známých pozorování

def ob_generator():
    new_pt_counter = 0
    for m in good: # cyklus přes korespondence
        kp1_id, kp2_id = m.queryIdx, m.trainIdx
        obs = self.observations

        # vyhledat pt3d_id aktuálního klíčového bodu z view1
        pt3d_id = obs[(obs.view_id == view1.index) & \
                      (obs.kp_id == kp1_id)].pt3d_id

        if pt3d_id.empty:
            # aktuální klíčový bod z view1 zatím není známý
            new_pt_counter += 1

            # generovat nové pt3d_id
            pt3d_id = handle_nan(obs.pt3d_id.max()) + new_pt_counter

            # vygenerovat záznam o pozorování z view1
            yield view1.index, kp1_id, pt3d_id
        else:
            # aktuální klíčový bod již je známý, použijeme jeho pt3d_id
            pt3d_id = pt3d_id.values[0]

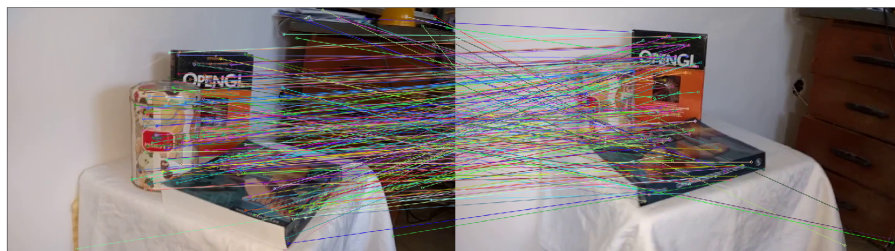
    # vygenerovat záznam o pozorování z view2
    yield view2.index, kp2_id, pt3d_id
```

Výpis 2.1: Generátor záznamů pozorování

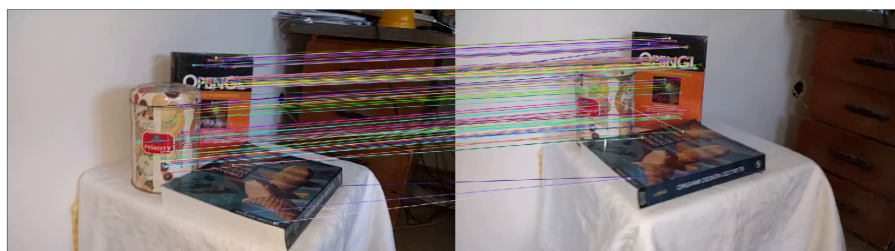


Z výpisu 2.1 je zřejmé že pro novou korespondenci již známého bodu vznikne jeden záznam, zatímco pro dosud neznámý bod vzniknou dva. V tuto chvíli můžeme o známých prostorových bodech prohlásit že existují a ve kterých pohledech jsou viditelné, ale nevíme nic o jejich prostorových souřadnicích.

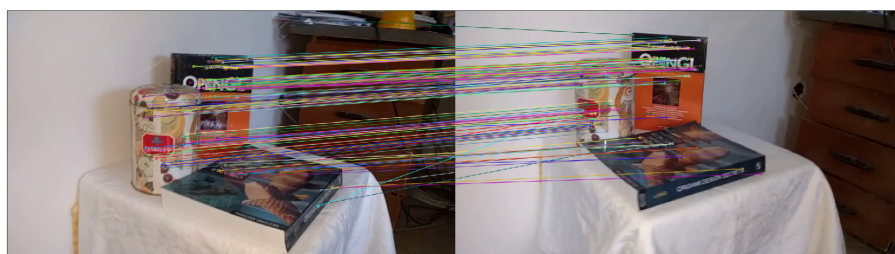
## Detekce outlierů



(a) Bez filtrace



(b) Po filtraci poměrem Minkovského vzdáleností deskriptorů



(c) Po filtraci maskou outlierů fundamentální matice

Obr. 2.1: Korespondující páry klíčových bodů před a po filtraci - jelikož pohyb kamery mezi snímky měl minimální rotační komponent, všechny přímky spojující příslušné korespondující body by měly být rovnoběžné. Různoběžnost přímky s ostatními indikuje pravděpodobnou falešnou shodu deskriptorů

## Poměr Minkowského vzdáleností deskriptorů

Tato metoda je již popsána v 1.2.1- Detekce falešných korespondencí pomocí poměru Minkowského vzdáleností, pozitivní důsledek na kvalitu korespondencí je zobrazen v Obr. 2.1b.

**Maska epipolární geometrie** O technice získání masky outlierů metodou RANSAC pojednává kapitola 1.3.2 - RANSAC metoda. Tuto techniku realizuje funkce knihovny *OpenCV* `cv2.findEssentialMat(pts1, pts2, imat, method=cv2.FM_RANSAC)` → E, mask.

Korespondující páry filtrované touto technikou jsou porovnány s nefiltrovanými v Obrázku 2.1c.

### Statistické výchylky v prostoru

Jako statistické kritérium využívám *Median Absolute Deviation* neboli *MAD*. *MAD* je pro množinu hodnot  $X = [X_1, X_2, \dots, X_n]$  definován jako

$$MAD(X) = \underset{i=1,2,\dots,n}{median}(|X_i - median(X)|) \quad (2.1)$$

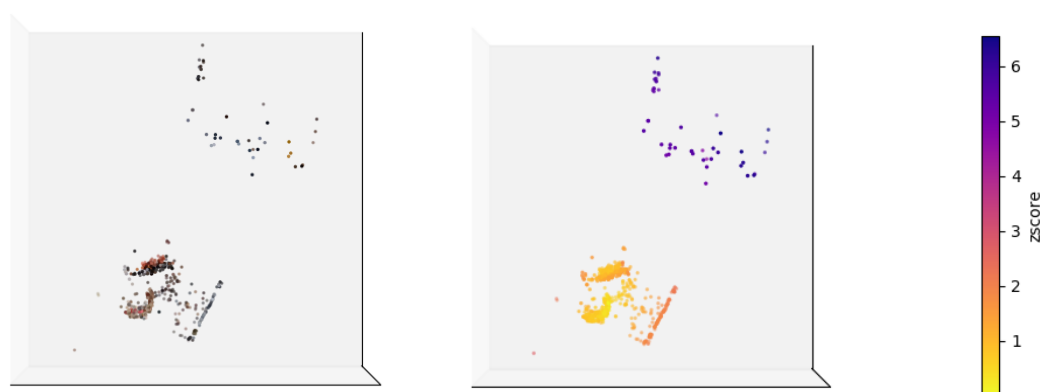
jemu příslušné standardní skóre (*zscore*) se vypočte:

$$zscore_i = 0.6745 \frac{X_i - median(X)}{MAD(X)} \quad (2.2)$$

$zscore_i$  udává, kolik modifikovaných standardních odchylek od středu množiny se bod  $X_i$  nachází.

Při triangulaci falešně korespondujícího páru bodů může dojít ke vzniku prostoro-  
vého bodu, jehož souřadnice se liší o několik řádů od souřadnic správných bodů. Pokud by bylo použité statistické kritérium založené na průměrování, pak by velmi vysoká odchylka výrazně ovlivnila střed zkoumané množiny.

Bod  $X_i$  je odstraněn, pokud jeho *zscore* překročí zadanou hodnotu. Odstranění provádí metoda `Reconstruction.remove_spatial_outliers(zthresh)`. Druhotným výhodným důsledkem této filtrace je odstranění bodů z pozadí scény, které nejsou součástí objektu zájmu.



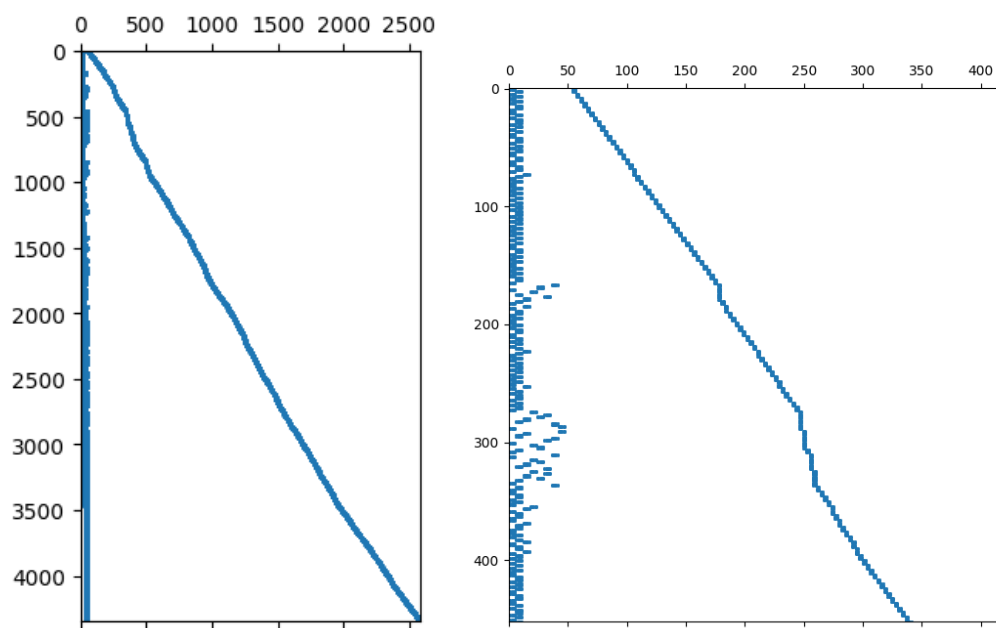
(a) Mrak bodů v původních barvách

(b) Mrak bodů zbarvený podle zscore

Obr. 2.2: Mrak bodů a jejich zscore - body s vyšším zscore budou odfiltrovány, v tomto případě se jedná o body mimo zájmovou oblast

## Bundle Adjustment

Bundle Adjustment je realizován třídou `BundleAdjuster`. Optimalizaci provádí funkce `least_squares` z knihovny `scipy.optimize`. Před použitím je nutné sestavit vektor optimalizovaných parametrů a matici řídkosti. Na to slouží metody `BundleAdjuster.bundle_update` a `BundleAdjuster.get_sparsity`. Velikost pracovních datových struktur je nastíněna v 1.3.4 - Bundle Adjustment.



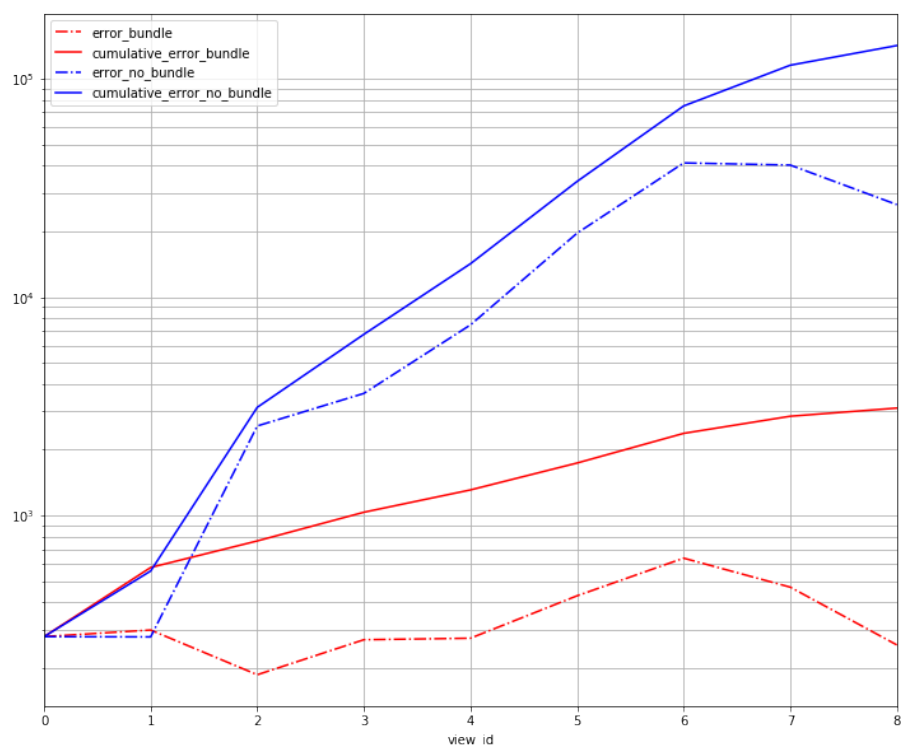
(a) Celá matice bundle adjustmentu

(b) Zvětšený segment matice 2.3a

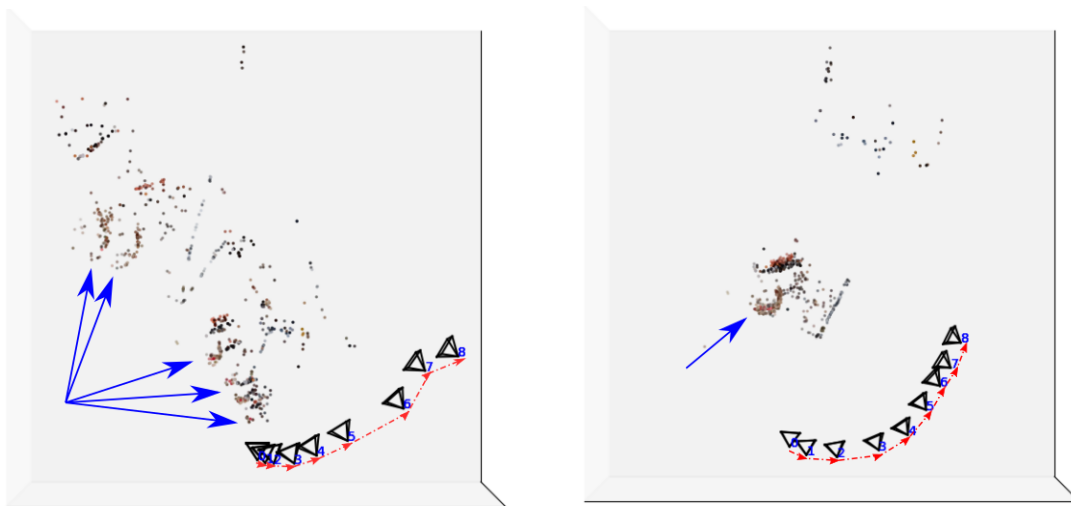
Obr. 2.3: Řídká matice bundle adjustmentu 2.3a sestává ze dvou jasně rozlišitelných částí - téměř vertikální linie v levé části reprezentuje vztahy mezi projekčními maticemi a prvky kritéria, a postupně se svažující pravá část reprezentuje vztahy mezi prostorovými body a prvky kritéria. Svahy těchto dvou částí jsou různé, protože bodů je mnohonásobně více než pohledů.

Důsledky bundle adjustmentu na mrak bodů získaný postupným skládáním 1.3.3 je zobrazen v obrázku 2.5a a 2.5b. Je patrné, že bundle adjustment dokáže zkorigovat všechny problémy jednoduchého postupného skládání zmíněné v 1.3.3

Vliv na reprojekční chyby konkrétních bodů je zobrazen v příloze A.



Obr. 2.4: Vliv bundle adjustmentu na součet reprojekčních chyb: rekonstrukce bez bundle adjustmentu má výslednou chybu o dva řády vyšší



(a) Mrak bodů získaný postupným skládáním (b) Mrak bodů 2.5a po bundle adjustmentu bez bundle adjustmentu

Obr. 2.5: Důsledek bundle adjustmentu na rekonstrukci scény: trajektorie kamery je vyznačena červeně, stejná struktura triangulovaná do různých míst je vyznačena svazkem modrých šipek

### 2.3.3 Vnitřní reprezentace dat

Informace o jednotlivých pozorováních klíčových bodů ve snímcích se uchovává v datovém rámci `Reconstruction.observations`. Každý řádek tohoto rámce podává informaci o jednom pozorování. Sloupec `view_id` uvádí index pohledu, `kp_id` uvádí index klíčového bodu v seznamu klíčových bodů příslušného pohledu a `pt3d_id` je index pozorovaného bodu.

```
>>> print(rec.observations.head(n=6))
```

	view_id	kp_id	pt3d_id
0	0	6	0
1	1	1	0
2	0	10	1
3	1	197	1
4	0	11	2
5	1	8	2

Výpis 2.2: Ukázka hlavičky datového rámce `Reconstruction.observations`

V ukázce 2.2 je vidět, že každému registrovanému prostorovému bodu musí odpovídat alespoň dvě pozorování z různých pohledů.

Prostorové souřadnice bodů se ukládají v datovém rámci `Reconstruction.pts3d`

```
>>> print(rec.pts3d.head(n=3))
```

	x	y	z
pt3d_id			
0	1.565949	-5.863913	14.087089
1	3.133807	-5.538301	13.706771
2	3.674357	-5.463591	13.416268

Výpis 2.3: Ukázka hlavičky datového rámce `Reconstruction.pts3d`

Informace o pozorování bodu a jeho faktická poloha jsou odděleny kvůli tomu, že je možné (dokonce nezbytné) rozhodovat o registraci bodu před tím, než je jeho poloha triangulována.

Barvy příslušící bodům jsou zjištěny z druhého ze dvou snímků použitých při triangulaci a ukládají se do rámce `Reconstruction.colors` v normalizované podobě.

```
>>> print(rec.colors.head(n=3))
```

	r	g	b
pt3d_id			
0	0.403922	0.419608	0.478431
1	0.509804	0.560784	0.627451
2	0.137255	0.200000	0.274510

Výpis 2.4: Ukázka hlavičky datového rámce `Reconstruction.colors`

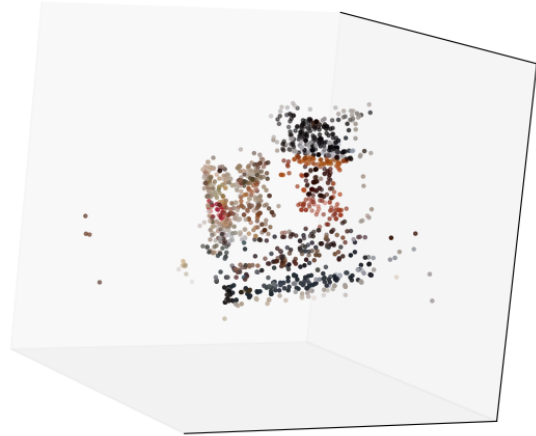
### 3 DEMONSTRAČNÍ SCÉNY

**Scéna s knihami** 3.1 je navržena tak aby její struktura byla v rekonstrukci rozpoznatelná. Záměrně jsem vybral dvě různobarevné knihy a válcovou plechovku, aby struktura scény byla dobře rozpoznatelná v rekonstrukci.

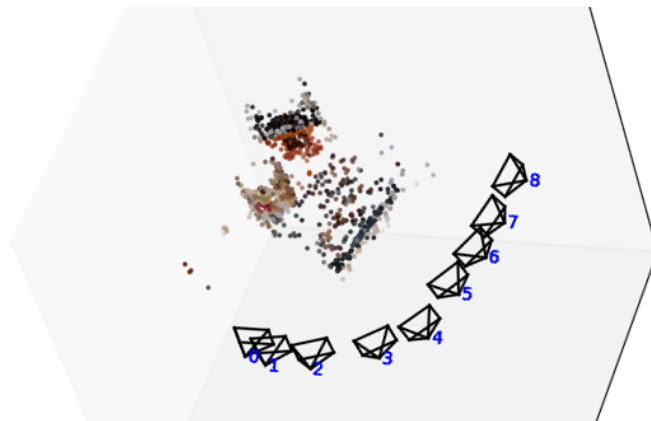
Samotná válcová nádoba je zobrazena ve větším rozmezí úhlů ve scéně 3.3.



(a) Snímek scény



(b) Rekonstruovaný mrak bodů



(c) Celá rekonstrukce včetně kamer

Obr. 3.1: Scéna s knihami a její mrak bodů (jeden z pohledů byl automaticky odstraněn během rekonstrukce, protože byl příliš blízko předchozímu)

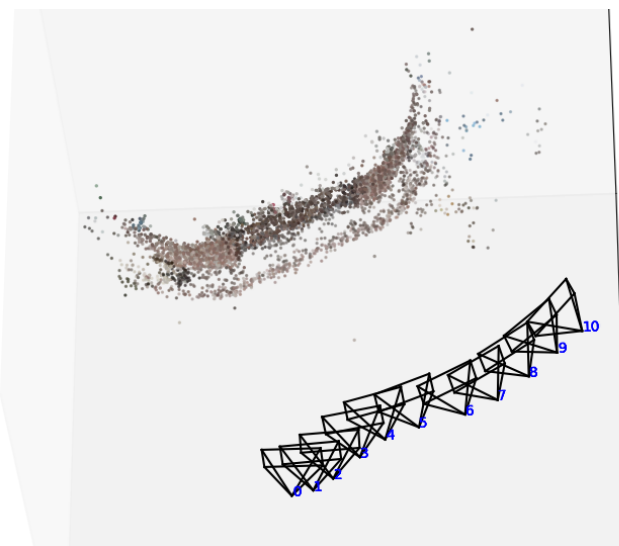
**Scéna s krbem** 3.2 je záběr cihlového krbu. Kamera se pohybuje po přímce a rotuje jen minimálně. Tato scéna dobře ilustruje vlastnosti detektorů klíčových bodů - drsné cihly a hlavně jejich hrany vyprodukují velké množství kvalitních klíčových bodů, zatímco bíle omítnutá část nevyprodukuje žádné. Pověšme si také téměř úplné ztráty jakékoliv informace o předmětech na římse kvůli nedostatku kvalitních klíčových bodů.



(a) Snímek scény



(b) Rekonstruovaný mrak bodů



(c) Celá rekonstrukce včetně kamer

Obr. 3.2: Scéna s krbem a její mrak bodů

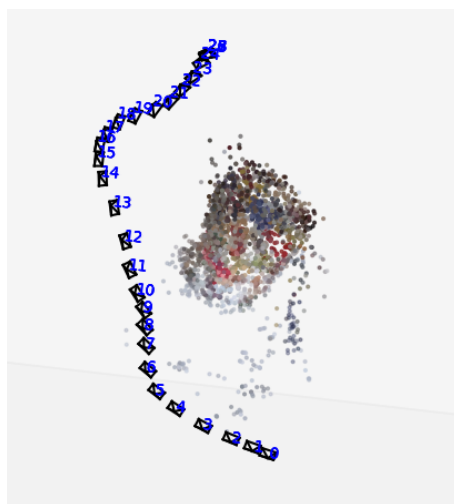
**Scéna s válcovou nádobou** 3.3 je třicetisnímkový záznam válcové nádoby ze scény s knihami. Větší množství snímků vede k většímu množství vychylujících bodů, což během optimalizace vede ke zkreslení celého mraku. V příloze B je srovnání průmětů tohoto modelu do původních pohledů.



(a) Snímek scény



(b) Rekonstruovaný mrak bodů



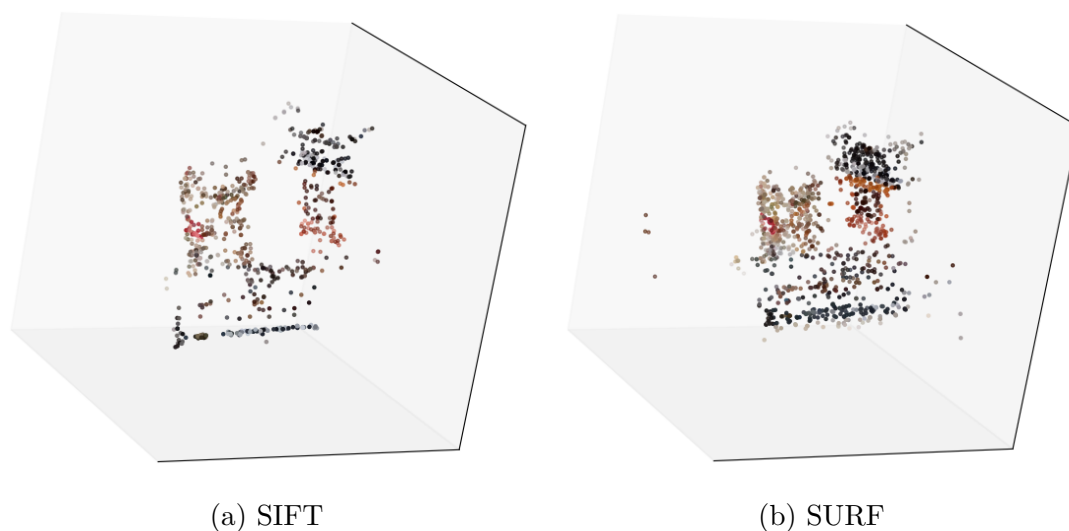
(c) Celá rekonstrukce včetně kamer

Obr. 3.3: Scéna s válcovou nádobou a její mrak bodů



## 4 ZÁVĚR

Můj program může s malými úpravami pracovat s jakýmkoli detektorem klíčových bodů, který odpovídá OpenCV API. Této skutečnosti jsem využil k porovnání algoritmů *SIFT* a *SURF*. Zaměřil jsem se na kvalitu výsledků, ne na rychlost výpočtu. Porovnání výsledných mraků bodů je v obrázku 4.1 a potvrzuje teoretický předpoklad o nadřazenosti *SURF*u z hlediska robustnosti klíčových bodů.



Obr. 4.1: Porovnání hustoty mraků získaných z různých druhů klíčových bodů: je zřejmé, že algoritmus *SURF* produkuje robustnější klíčové body a tedy i hustší mrak

Navržený algoritmus mě překvapil kvalitou výsledných mraků bodů, obzvláště pokud zvážím jeho jednoduchost. Hlavní příčinou jeho efektivity je využití formulace rekonstrukce jako optimalizační úlohy (1.3.4). Tento krok je natolik zásadní (2.4, A), že přináší nový úhel pohledu na celou úlohu. Můj první instinkt byl uvažovat o tvorbě modelu jako o sérii dvoupohledových rekonstrukcí, které jsou nakonec "zahlazený" bundle adjustmentem. Nyní si však myslím, že je vhodnější uvažovat o celkové rekonstrukci ve smyslu formulovaném v 1.3.4. Ostatní kroky (výpočet epipolární geometrie, triangulace) slouží pouze jako nástroj k vytvoření rozumných počátečních stavů pro algoritmus nelineární optimalizace.

Je vhodné také poukázat na skutečnost, že optimalizační proces může vykonávat i více funkcí než bylo využito v řešení této práce. Je například možné zahrnout intrinsické parametry nebo distorční koeficienty kamery ve vektoru optimalizovaných proměnných. Tímto způsobem by mohlo být možné zbavit se závislosti na znalosti intrinsických parametrů předem - stačil by odhad založený například na velikosti snímků. Cenou za rozšíření vektoru proměnných je samozřejmě zvýšení rozměru úlohy a tedy i výpočetních nároků. Praktická implementace programu je provedena

v angličtině (názvy proměnných a komentáře) kvůli použití stejného názvosloví jako v použité literatuře a pro konzistenci s jazykem knihoven. Zdrojový text je segmentován do funkčních bloků, které odpovídají dílčím úlohám popsaným v teoretické části a obsahuje i nástroje vhodné pro inspekci a vizualizaci zpracovávaných dat (2.2). Volba nástrojů (2.1) byla vesměs vhodná, s výjimkou knihovny *matplotlib* pro vizualizaci 3d mraku bodů. Knihovna *matplotlib* je navržena hlavně pro práci s 2d grafy a tuto roli plní velmi dobře, ale její 3d nadstavba má neintuitivní API a chybí ji některé důležité prvky pro ovládání 3d zobrazení. Jako náhradu by bylo možné využít knihovnu *mayavi*, která je zaměřená specificky na 3d vizualizace a je z velké části kompatibilní se *SciPy Stackem*.

Možná vylepšení samotného algoritmu by se měla vydat jedním ze dvou směrů:

1. Řešení nedokonalostí výsledných mraků, které již byly zmíněny v popisech scén:

- Zhuštění mraku bodů
- Snížení počtu falešných shod bodů

Tyto dva problémy spolu souvisí skrze robustnost klíčových bodů - zvýšení nároků na kvalitní shody povede k menšímu celkovému počtu bodů a snížení nároků povede k většímu počtu falešných shod.

2. Odpoutání se od předpokladu zavedených v úvodu:

- Nezávislost na seřazení snímků  
možné řešení: využít grafovou strukturu pro organizaci dat
- Autokalibrace tj. odvození intrinsických parametrů bez předchozí znalosti  
možné řešení: odhadnout, poté optimalizovat během bundle adjustmentu
- Zobecnění vstupních dat  
možné řešení: zvětšit množinu vstupních snímků a zvýšit nároky na kvalitní klíčové body a shody, potom odstranit snímky, které jich nemají dostatečné množství

# LITERATURA

- [1] LOWE, David G.; *Distinctive Image Features from Scale-Invariant Keypoints* Computer Science Department, University of British Columbia, Vancouver, B.C., Canada
- [2] BAY, Herbert; ESS, Andreas; TUYTELAARS, Tinne; VAN GOOL, Luc; *Speeded-Up Robust Features(SURF)* ETH Zurich, BIWI; K. U. Leuven, ESAT-PSI
- [3] FISHER, Robert B.; *Sub-pixel estimation* University of Edinburgh
- [4] WITKIN, Andrew P.; *Scale-space Filtering* Fairchild Laboratory for Artificial Intelligence Research
- [5] SHAH, Mubarak; *Lecture 05 - Scale-invariant Feature Transform (SIFT)* UCF Computer Vision Video Lectures 2012  
<https://www.youtube.com/watch?v=NPcMS49V5hg>
- [6] SIMEK, Kyle; *Dissecting the Camera Matrix, Part 3: The Intrinsic Matrix* Sightations ← A Computer Vision Blog <https://ksimek.github.io/2013/08/13/intrinsic/>
- [7] COLLINS, Robert T.; *Lecture 06: Harris Corner Detector* Pennsylvania State University, Introduction to Computer Vision  
<http://www.cse.psu.edu/~rtc12/CSE486/>
- [8] HARTLEY, Richard; ZISSERMAN, Andrew; *Multiple View Geometry in Computer Vision* Cambridge University Press [isbn-13: 978-0-521-54051-3]
- [9] HORÁK, Karel; *Přednáška 3 - Diskrétní obraz* FEKT VUT Brno, Zpracování vícerozměrných signálů  
[http://midas.uamt.feec.vutbr.cz/ZVS/zvs\\_cz.php](http://midas.uamt.feec.vutbr.cz/ZVS/zvs_cz.php)
- [10] HARTLEY, Richard I.; STURM, Peter; *Triangulation* GE-CRD, Schenectady, NY; Lifa-Inria, Grenoble, France
- [11] OpenCV Documentation *Introduction to SURF (Speeded-Up Robust Features)*  
[http://docs.opencv.org/3.0-beta/doc/py\\_tutorials/py\\_feature2d/py\\_surf\\_intro/py\\_surf\\_intro.html](http://docs.opencv.org/3.0-beta/doc/py_tutorials/py_feature2d/py_surf_intro/py_surf_intro.html)
- [12] PEDERSEN, Jacob Toft; *Study group - SURF: Feature detection & description*  
<http://cs.au.dk/~jtp/>

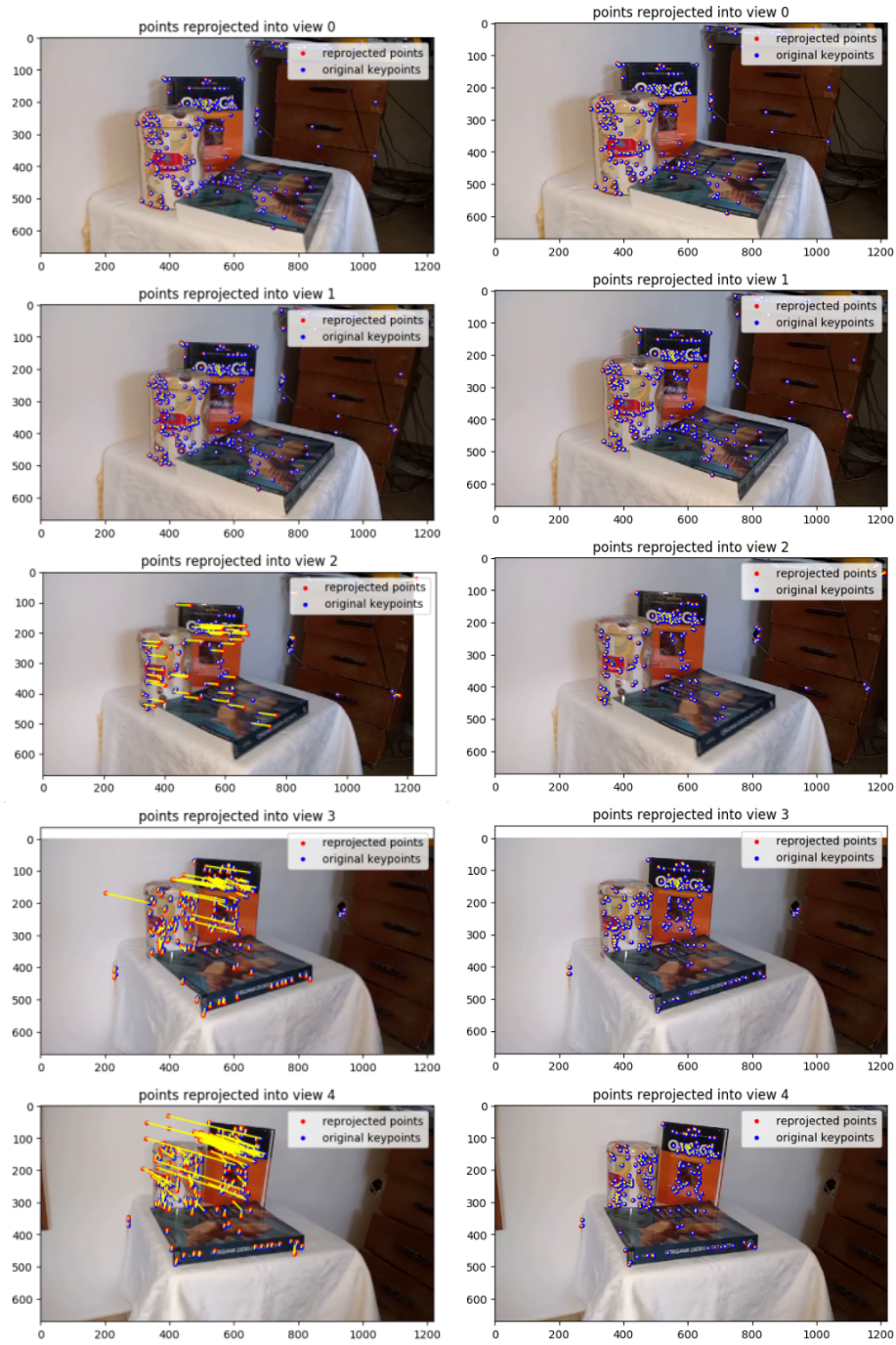
- [13] FILLIEN, James J.; Heckert Alan *Engineering Statistics Handbook - Exploratory Data Analysis* <http://www.itl.nist.gov/div898/handbook/eda/eda.htm>
- [14] Stachniss, Cyrill; *Photogrammetry II Video lectures* University of Bonn, Germany <https://www.youtube.com/playlist?list=PLgnQpQtFTOGRsi5vzy9PiQpNWHjq-bKN1>
- [15] Bird, Nate; *Camera Representation Part3* <http://natebird.com/tag/camera-representation/>
- [16] BEAUCHEMIN, S. S.; BAJCSY, R.; GIAVATY, G.; *Modelling and Removing Radial and Tangential Distortions in Spherical Lenses* GRASP Laboratory, Department of Computer and Information Science, University of Pennsylvania
- [17] NEUBAUER, Wolfgang; *Laser Scanning and Archaeology* VIAS-Vienna Institute for Archaeological Science, Geophysical Prospection, Geodesy and Photogrammetry, University of Vienna, Austria <https://www.gim-international.com/content/article/laser-scanning-and-archaeology>
- [18] Bin-Yih Juang; Chiou-Shann Fuh; *3-D Terrain Reconstruction with Aerial Photography* National Taiwan University, Taipei, Taiwan
- [19] AGARWAL, Sameer; SNAVELY, Noah; SIMON, Ian; SEITZ, Steven M.; SZELISKI, Richard; *Building Rome in a Day* University of Washington; Cornell University; Microsoft Research <https://grail.cs.washington.edu/rome/>
- [20] *Rovina project* Department of Photogrammetry University of Bonn <http://www.rovina-project.eu/>
- [21] MAYOROV, Nikolay; *Trust Region Reflective Algorithm* <https://nmayorov.wordpress.com/2015/06/19/trust-region-reflective-algorithm/>
- [22] TOMASI, Carlo; *Vector Representation of Rotations*

## SEZNAM PŘÍLOH

A	Vliv bundle adjustmentu na reprojekční chyby	50
B	Reprojekce válcové nádoby do původních pohledů	51
C	Rotační vektor a matice	52
D	Obsah DVD	53



# A VLIV BUNDLE ADJUSTMENTU NA REPROJEKČNÍ CHYBY



(a) Reprojekční chyby bez bundle adjustmentu (b) Reprojekční chyby po bundle adjustmentu

Obr. A.1: Důsledek bundle adjustmentu na rekonstrukci scény: rozdíl poloh původních klíčových bodů a jejich reprojekcí je vyznačen žlutě. Pohledy odpovídají mraku bodů na Obr. 2.5a a 2.5b

## B REPROJEKCE VÁLCOVÉ NÁDOBY DO PŮ- VODNÍCH POHLEDŮ



Obr. B.1: Reprojekce válcové nádoby do původních pohledů



## C ROTAČNÍ VEKTOR A MATICE

Pro údaj o rotaci ve třech rozměrech se v této práci využívají dva různé způsoby reprezentace:

1. Rotační matice
2. Rotační vektor

Rotační matice se využívá v projekční matici a je výhodnější z hlediska přehlednosti výpočtů, protože ji lze aplikovat maticovým násobením. Např. rotace vektoru  $\mathbf{v} = (1, 0, 0)$  o  $45^\circ$  kolem osy  $z$ :

$$\hat{\mathbf{v}} = \begin{bmatrix} \cos(45^\circ) & -\sin(45^\circ) & 0 \\ \sin(45^\circ) & \cos(45^\circ) & 0 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{v} = \begin{bmatrix} \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} & 0 \\ \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} \frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} \\ 0 \end{bmatrix} \quad (\text{C.1})$$

Rotační matice je méně efektivní datovou strukturou, jelikož ukládá tři stupně volnosti v devíti parametrech.

Druhou reprezentací je rotační vektor ve formě součinu úhlu rotace jednotkového vektoru jehož směr udává osu rotace.  $\mathbf{r} = \theta \mathbf{u}$ . Rotaci  $\mathbf{v}$  pomocí takového vektoru lze realizovat Rodriguesovým vzorcem:

$$\theta = |\mathbf{r}| \quad (\text{C.2})$$

$$\mathbf{u} = \frac{\mathbf{r}}{\theta} \quad (\text{C.3})$$

$$\hat{\mathbf{v}} = \mathbf{v} \cdot \sin(\theta) + (\mathbf{u} \times \mathbf{v}) + \mathbf{u}(\mathbf{u} \cdot \mathbf{v})(1 - \sin(\theta)) \quad (\text{C.4})$$

Vztah mezi rotační maticí a rotačním vektorem:

$$\mathbf{R} = \cos(\theta)\mathbf{I} + (1 - \cos(\theta)) \cdot \mathbf{u}\mathbf{u}^T + \sin(\theta) \cdot [\mathbf{u}]_{\times} \quad (\text{C.5})$$

[22]

## D OBSAH DVD

Příložené DVD obsahuje kromě textu práce a programu také vstupní soubory pro demonstrační scény, skript pro kalibraci kamery a soubor se záznamem šachovnice pořízený stejnou kamerou jako záznamy scén.

```
DVD
├── debug
├── input_data
│   ├── camera_calibration
│   │   ├── nexus5x
│   │   │   └── nexus5x_720p.mp4
│   │   ├── calibrate.py
│   │   └── calibrate_video.py
│   ├── image_sequences
│   └── video
│       ├── knihy.mp4
│       ├── krb.mp4
│       └── nadoba.mp4
├── __init__.py
├── bundle_adjustment.py
├── reconstruction.py
├── sladecek_martin_3dmodel.pdf
├── software.txt
└── utils.py
```